

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : **09-265397**

(43)Date of publication of application : **07.10.1997**

(51)Int.Cl.

G06F 9/38  
G06F 9/38

(21)Application number : **08-075513**

(71)Applicant : **HITACHI LTD**

(22)Date of filing : **29.03.1996**

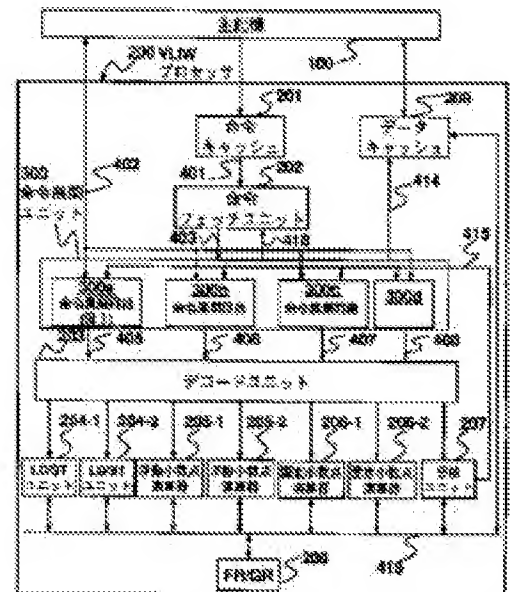
(72)Inventor : **TSUSHIMA YUJI**  
**TANAKA GIICHI**  
**TAMAOKI YOSHIKO**  
**ITO MASANAO**  
**SHIMADA KENTARO**  
**TOTSUKA YONETARO**  
**NAGASHIMA SHIGEO**

## (54) PROCESSOR FOR VLIW INSTRUCTION

### (57)Abstract:

PROBLEM TO BE SOLVED: To reduce NOP(no-operation) instructions and to execute a VLIW(very long instruction word) instruction whose length is shortened.

SOLUTION: The number of the succeeding NOP instructions of the small instruction is added for each small instruction in a long instruction and the NOP instructions are eliminated from the succeeding long instruction. Thereafter, the plural small instructions in the respective long instructions are divided into plural groups, the operation codes are replaced with group codes for which the combination of the operation code of the small instructions of the respective groups is compressed and compressed group instructions are prepared. An instruction development unit 300 is provided with instruction development circuits 300a-300d for the respective group instructions and each instruction development circuit, 300a for instance, develops one group instruction in the long instruction, generates a group of the small instructions specified by the instruction and supplies them through a decoding unit 203 to a function unit 204-1 or the like. At the time, the NOP instructions for



the number specified by an NOP number for the respective small instructions present in the group instruction are supplied after the respective small instructions.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-265397

(43)公開日 平成9年(1997)10月7日

(51)Int.Cl. <sup>6</sup>	類別記号	片内整理番号	P I	技術表示箇所
G 0 6 F 9/38	3 7 0		G 0 6 F 9/38	3 7 0 B
	3 1 0			3 1 0 H

審査請求 未請求 請求項の数14 OL (全 29 頁)

(21)出願番号	特願平8-75513	(71)出願人	000005108 株式会社日立製作所 東京都千代田区神田駿河台四丁目6番地
(22)出願日	平成8年(1996)3月29日	(72)発明者	對島 雄次 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内
		(72)発明者	田中 義一 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内
		(72)発明者	玉置 由子 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内
		(74)代理人	弁理士 藤田 利幸

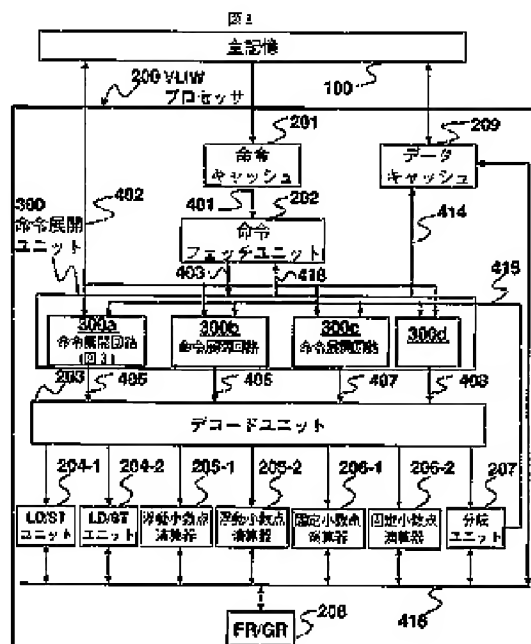
最終頁に続く

(54) 【発明の名称】 VLIW命令用プロセッサ

(57)【要約】

【課題】NOP命令を減らし、長さを短めたVLIW命令を実行可能にする。

【解決手段】長命令中の小命令毎にその小命令の後続のNOP命令の数を付加し、後続の長命令からはそれらのNOP命令を削除する。その後各長命令中の複数の小命令を複数のグループに分け、各グループの小命令のオペコードの組み合わせを圧縮したグループコードでもってそれらのオペコードを置き換えて圧縮されたグループ命令を作る。命令展開ユニット300は、各グループ命令用の命令展開回路300aから300dを有し、各命令展開回路、例えば、300aは、長命令中の一つのグループ命令を展開してその命令が指定する一群の小命令を生成し、デコードユニット203を介して、機能ユニット204-1等に供給する。このとき、このグループ命令内にあるそれぞれの小命令に対するNOP数で指定される数だけのNOP命令を、それぞれの小命令の後に供給する。



## 【特許請求の範囲】

【請求項 1】 実行すべきプログラムを構成する複数の圧縮された長命令を順次供給する長命令供給回路と、

該長命令供給回路により供給された一つの圧縮された長命令を展開して複数の小命令を生成する命令展開ユニットと、

該複数の小命令を並列に実行するための複数の機能ユニットとを有し、

該複数の機能ユニットは複数の機能ユニット群に区別され、

該命令展開ユニットは、それぞれ該複数の機能ユニット群の一つに対応して設けられた、互いに並列に動作する複数の命令展開回路を有し、

各圧縮された長命令は、該複数の機能ユニット群の一つにそれぞれ対応した複数の命令フィールドを有し、各命令フィールドは、その命令フィールドに対応する機能ユニット群により並列に実行されるべき一群の小命令を該一群の小命令よりも少ない情報をもって表す圧縮されたグループ命令を含み、

各命令展開回路は、該命令供給ユニットにより供給された一つの長命令内の、その各命令展開回路に対応する一つの機能ユニット群に対応する一つの命令フィールドに含まれた圧縮されたグループ命令を展開して、そのグループ命令を表す一群の小命令を生成し、生成された一群の小命令を上記対応する一つの機能ユニット群に並列に出力する V L I W 命令用プロセッサ。

【請求項 2】 各長命令の各命令フィールドに含まれた圧縮されたグループ命令は、そのグループ命令により表される一群の小命令の一つがそれぞれ必要とする一群のオペコードを、それらの一群のオペコードよりも少ない情報で表す圧縮されたグループコードと、それぞれ該一群の小命令の一つが必要とする少なくとも一つのオペランドを表す一群のオペランド情報とを含み、

各命令展開回路は、

該命令供給ユニットにより供給された上記一つの長命令内の、その命令展開回路に対応する命令フィールドに含まれた圧縮されたグループコードを展開してそのグループコードにより表される一群の小命令が必要とする一群のオペコードを生成して並列に出力するグループコード展開回路と、

該出力された一群のオペコードの各々に上記一群のオペランド情報の一つを結合して一群の小命令を生成し、生成された一群の小命令を、その各命令展開回路に対応する上記一つの機能ユニット群に並列に出力する小命令生成ユニットとを有する請求項 1 記載の V L I W 命令用プロセッサ。

【請求項 3】 各長命令の各命令フィールドの上記圧縮されたグループコードは、あらかじめ定められた複数の命令テーブル内の一つの命令テーブル内の一つのエントリを指定するポインタとなるように定められ、

ここで、該複数の命令テーブルは、複数の群のオペコードを登録するテーブルであり、

各命令テーブルは、いずれか一つの機能ユニット群に対応する、上記複数の長命令内の複数の命令フィールドに対応して定められた、

各命令テーブルに登録された上記複数の群のオペコードは、上記複数の長命令に含まれ、いずれもその各命令テーブルに対応する複数の命令フィールド内の圧縮された複数のグループコードにより表される複数の群のオペコードの内、異なる組み合わせを有する複数の群のオペコードであり、

上記一つの命令テーブルは、上記複数の命令テーブルの内、上記各長命令の上記各命令フィールドに対応する上記一つの機能ユニット群に対応する命令テーブルであり、

上記一つのエントリは、上記各長命令内の上記各命令フィールド内の上記圧縮されたグループコードを表す一群のオペコードが登録されるエントリである請求項 2 記載の V L I W 命令用プロセッサ。

【請求項 4】 各命令展開回路内の該グループコード展開回路は、複数のグループコードを表す複数の群のオペコードを保持し、上記長命令供給回路から供給されたいずれかのグループコードに応答して、該複数の群のオペコードの内、該供給されたグループコードを表す一群のオペコードを出力するメモリを有し、

該メモリに保持された該複数の群のオペコードは、上記複数の命令テーブルの一つに登録された複数の群のオペコードであり、

その一つの命令テーブルは、上記複数の命令テーブルの内、その各命令展開回路に対応する一つの機能ユニット群に対応する、上記複数の長命令内の複数の命令フィールドに対応して設けられた命令テーブルである請求項 3 記載の V L I W 命令用プロセッサ。

【請求項 5】 上記複数の命令テーブルは、該プロセッサに接続されるべき主記憶に保持され、

上記 V L I W 命令用プロセッサは、該プログラム内に含まれた一つの特定のロード命令に応答して、該主記憶内のその特定のロード命令が指定する記憶位置に保持された一つの命令テーブルに属すべき複数の群のオペコード

を、該複数の命令展開回路の一つに含まれた上記メモリに読み出す回路を更に有する請求項 4 記載の V L I W 命令用プロセッサ。

【請求項 6】 上記複数の各命令展開回路内の内の少なくとも一つに含まれた上記グループコード展開回路内の上記メモリは、その一つの命令展開回路に対応する一つの機能ユニット群に対応する、上記複数の長命令内の一つの命令フィールドに対応して定められた一つの命令テーブルに含まれた複数の群のオペコードを全て保持するに必要な容量を有する請求項 4 記載の V L I W 命令用プロセッサ。

【請求項7】上記複数の命令テーブルの少なくとも一つは、上記プログラムの複数の部分プログラムに対応して定められた複数の部分テーブルからなり、該複数の部分テーブルは該主記憶に保持され、上記VLIW命令用プロセッサは、上記プログラムの各部分プログラムの実行開始前に、上記複数の各命令展開回路内の内の少なくとも一つに含まれた上記メモリに上記主記憶から該複数の部分テーブルの一つに含まれた複数群のオペコードを読み出す回路をさらに有する請求項4記載のVLIW命令用プロセッサ。

【請求項8】上記複数の命令展開回路の少なくとも一つに含まれた上記グループコード展開回路は、上記命令供給回路からその一つの命令展開回路に供給されたグループコードに対応する一群の命令コードがそのグループコード展開回路内の上記メモリに保持されているか否かを検出する回路と、

その対応する一群のオペコードがそのメモリに保持されていないと検出されたときには、該主記憶より該一群のオペコードを読み出し、該メモリに保持する回路とを有する請求項5記載のVLIW命令用プロセッサ。

【請求項9】上記複数の命令展開回路の少なくとも一つに含まれた上記グループコード展開回路内の上記メモリは、

予め定めた複数のグループコードに対応する複数群のオペコードを保持する第1のメモリと、

上記予め定めた複数のグループコード以外の複数のグループコードに対応する他の複数群のオペコードを保持するための第2のメモリとを有し、

上記VLIW命令用プロセッサは、

上記主記憶から複数群のオペコードを上記第2のメモリに読み出す回路と、

上記プログラムの実行過程で異なる複数群のオペコードを読み出すように、上記読み出し回路による読み出しを指示する回路とを有する請求項4記載のVLIW命令用プロセッサ。

【請求項10】各長命令の各命令フィールドに含まれた圧縮されたグループ命令は、そのグループ命令内の圧縮されたグループコードが表す一群の小命令に対応する一群のNOP数を更に有し、

各NOP数は、該一群の小命令内のその各NOP数に対応する一つの小命令を実行した後に、その対応する小命令を実行する一つの機能ユニットにより実行すべきNOP命令の数を表し、

各命令展開回路内の上記小命令生成ユニットは、その各命令展開回路に対応する機能ユニット群内の複数の機能ユニットに対応して設けられた一群の小命令生成回路を有し、

各小命令生成回路は、

上記長命令供給回路から供給された一つの長命令内の一つのグループ命令の圧縮されたグループコードに対して

その各命令展開回路内の上記グループコード展開回路により生成された一群のオペコードの一つと、その一つのグループ命令内の一群のオペランド情報の一つとを結合して、一つの小命令を生成し、その各小命令生成回路に対応する一つの機能ユニットに供給し、その供給の後に、上記一つのグループ命令に含まれる、一群のNOP数の一つにより指定される数のNOP命令を順次その小命令生成回路に対応する機能ユニットに供給する選択回路と、

上記選択回路が上記各命令展開回路から上記一群のオペコードの後に供給される、上記一つの長命令の後続の長命令に対する後続の一群のオペコードの一つを含む小命令を上記対応する機能ユニットに供給するのを、上記一つのNOP数で指定される数のNOP命令の供給が完了するまで禁止する制御回路とを有する請求項2記載のVLIW命令用プロセッサ。

【請求項11】各命令展開回路は、その各命令展開回路に対応する機能ユニット群内の複数の機能ユニットに対応して設けられた複数のオペコードキューをさらに有し、

各オペコードキューは、上記グループコード展開部と上記一群の小命令生成回路の一つに接続され、該長命令供給回路から該各命令展開回路に供給された上記一つの圧縮された長命令の一つのグループ命令のグループコードに対して、その各命令展開回路内の上記グループコード展開回路から一群のオペコードが出力されるとに、その一群のオペコードの一つを保持するように、該グループコード展開回路から供給される複数のオペコードを順次保持し、保持された複数のオペコードをそれらの保持順に順次、そのオペコードキューに接続された上記一つの小命令生成回路に出力し、

各小命令生成回路内の上記制御回路は、上記複数のオペコードキューの内、その各小命令生成回路に接続された一つのオペコードキューに対して、上記一つのNOP数で指定される数のNOP命令の供給が完了するのに同期して、次のオペコードを出力するように、上記その各小命令生成回路に接続された一つのオペコードキューに指示する回路を有する請求項10記載のVLIW命令用プロセッサ。

【請求項12】各命令展開回路は、上記長命令供給回路と上記グループコード展開回路に接続されたグループコードキューと、

上記長命令供給回路とその各命令展開回路内の上記一群の小命令生成回路の一つにそれぞれ接続された一群のオペランドキューと、

その各命令展開回路内の上記一群の小命令生成回路の一つにそれぞれ接続された一群のNOP数キューとをさらに有し、

上記グループコードキューは、それぞれ上記長命令供給回路より順次供給される複数の長命令内の、それぞれそ

の各命令展開回路に対応する機能ユニット群に対応する複数の命令フィールドの一つにそれぞれ含まれる複数の圧縮されたグループコードを保持し、保持された複数の圧縮されたグループコードをそれらの保持順に順次上記グループコード展開回路へ出力し、

上記一群のオペランドキューの各々は、上記複数の長命令内の、上記複数の命令フィールドにそれぞれ含まれた複数のオペランド情報の内、その各命令展開回路に対応する上記機能ユニット群の一つにより実行されるべき複数の小命令により使用される一群のオペランド情報を保持し、その各オペランドキューに接続された小命令生成回路にそれらの保持されたオペランド情報をそれらの保持順に順次供給し、

上記一群のNOP数キューの各々は、上記複数の長命令内の、上記複数の命令フィールドにそれぞれ含まれた複数のNOP数の内、その各命令展開回路に対応する上記機能ユニット群の一つにより実行されるべき複数の小命令により使用される一群のNOP数を保持し、その各NOP数キューに接続された小命令生成回路にそれらの保持されたNOP数をそれらの保持順に順次供給し、  
上記命令展開回路内の上記一群の小命令生成回路の各々内の上記制御回路は、上記各小命令生成回路内の選択回路により供給すべきひとつ又は複数のNOP命令の供給が完了するのに同期して、上記一群のオペランドキューの内のその各小命令生成回路に接続された一つのオペランドキューと上記一群のNOP数キューの内のその各小命令生成回路に接続された一つのNOP数キューとに、次のオペランドおよび次のNOP数の供給要求を出力する回路を有する請求項1記載のVLIW命令用プロセッサ。

【請求項13】各命令展開回路は、そこに含まれた上記一群の小命令生成回路の一つにそれぞれ含まれる複数の制御回路に接続され、上記複数の制御回路のいずれかから出力された、上記次のオペランドおよび次のNOP数の供給要求に responding、上記グループコードキューに次のグループコードキューを上記グループコード展開部に供給することを要求する回路をさらに有する請求項12記載のVLIW命令用プロセッサ。

【請求項14】実行すべきプログラムを構成する複数の長命令を順次供給する長命令供給回路と、  
複数の小命令を並列に実行するための複数の機能ユニットと、

該長命令供給回路により供給された長命令により指定される複数の小命令を該複数の機能ユニットに供給する命令供給ユニットとを有し、

各長命令は、それぞれ該複数の機能ユニットの一つに対応した複数の命令フィールドを有し、各命令フィールドは、少なくとも一つの小命令とその小命令に対応するNOP数を指定する情報を有し、  
該NOP数は、該一つの小命令をその各命令フィールド

に対応する機能ユニットにより実行した後、その対応する機能ユニットにより実行すべきNOP命令の数を表し、

上記命令供給ユニットは、該複数の機能ユニットの一つにそれぞれ対応して設けられた一群の小命令生成回路を有し、

各小命令生成回路は、

上記長命令供給回路から供給された一つの長命令内の一つの命令フィールドに含まれた情報により指定される一つの小命令をその各小命令生成回路に対応する一つの機能ユニットに供給し、その供給の後に、上記一つの命令フィールドに含まれる上記情報で指定されるNOP数に等しい数のNOP命令を順次その対応する機能ユニットに供給する選択回路と、

上記選択回路が、上記各長命令供給回路から供給される後続の長命令に含まれた小命令を上記対応する機能ユニットに供給するのを、上記一つのNOP数に等しい数のNOP命令の供給が完了するまで禁止する制御回路とを有するVLIW命令用プロセッサ。

【発明の詳細な説明】

【0001】

【発明の属する技術の分野】本発明は、VLIW (VERY LONG INSTRUCTION WORD) 命令（以下では長命令とも呼ぶ）を展開して実行するプロセッサに関する。

【0002】

【従来の技術】計算機の性能は、マシンサイクルと、1命令の実行に必要なマシンサイクル数を表す、命令当たりのサイクル数CPI (CYCLES PER INSTRUCTION) により決定される。計算機の性能向上のためには、マシンサイクル、CPIを共に小さくすることが肝要となる。CPIを小さくするための方式には、1マシンサイクルで同時に多数の命令を処理する方式がある。この方式の代表的な例の一つにVLIW方式がある（ヘネシー&パターソン「コンピュータ・アーキテクチャ」参照）。

【0003】VLIW方式では、複数の命令フィールドからなる長い命令を使用し、個々の命令フィールドが演算器や記憶装置等の機能ユニットを制御する。このために1命令で複数の機能ユニットを制御出来る。命令発行回路を単純化するために、VLIW命令の各命令フィールドは特定の命令（以下これを小命令と呼ぶ）に割り当てられている。同一のVLIW命令内の複数の小命令は、それぞれに対応する複数の機能ユニットを同時に制御可能である。各小命令は、演算を示すオペコードと演算の対象を表すオペランドにより構成されている。VLIW方式では、コンパイル時に、プログラム中の小命令の依存関係を考慮して、同時に実行できる小命令が同一のVLIWに出来るだけ多く含まれるように、プログラム中の小命令の実行順序をスケジュールし、複数のVLIW命令中に配置する。このため、各VLIW命令内の

多数の小命令は並列に実行できるので、このような命令を実行する計算機は複雑な命令発行回路を必要としない。これにより、マシンサイクルの短縮や同時に発行できる命令数（以下では、命令並列度と呼ぶ）を上げ、命令当たりのサイクル数CPIを下げるのが容易となり、計算機の性能向上技術として注目されている。

【0004】VLIW方式では、各VLIW命令内に、各機能ユニットに対応した命令フィールドを有するため、そのVLIW命令で使用しない機能ユニットがある場合には、その機能ユニットに対応した命令フィールドに、何の動作もしないことを指定するNOP（NO OPERATION）命令を配置する。このため、プログラムによっては、多くのVLIW命令内に多くのNOP命令が埋め込まれることが生じる。VLIW命令内の多くの命令フィールドにNOP命令が埋め込まれると、プログラムを構成するVLIW命令の数が増大し、これらの命令を保持するために主記憶や命令キャッシュが浪費されることになる。

【0005】NOP命令の削減に関していくつかの提案がなされている。たとえば、情報処理学会研究報告、93-ARC-102号、第17ページから第24ページ（以下、第1の参考文献と呼ぶ）では、NOP命令のみからなる一つ又は複数の連続する無効なVLIW命令を除去するために、このような一つ又は複数のVLIW命令でもって表現すべき遅延サイクル数を保持するフィールドを、その一つまたは複数のVLIW命令の直前に実行すべき先行する有効なVLIW命令中に設け、その先行する有効なVLIW命令を実行後、その遅延サイクル数経過した後に、その一つ又は複数のVLIW命令の後に実行すべきであった後続の有効なVLIW命令を実行するようにしている。この技術ではVLIW命令数を減少することが出来るので、この技術はVLIW命令を時間的に圧縮する方法とも言える。さらに、この従来技術では、全てのフィールドがNOP命令であるようなVLIW命令が連続している場合、別の命令流に切り替えるマルチスレッド処理により機能ユニットの使用効率が高められる方式が提案されている。

【0006】情報処理学会研究報告、94-ARC-107号、第113ページから第120ページ（以下、第2の参考文献と呼ぶ）あるいは情報処理学会、「並列処理シンポジウムJSP'92」論文集、第265頁から第272頁（以下、第3の参考文献と呼ぶ）では、VLIWを時間的に圧縮する他の方法が提案されている。すなわち、各VLIW命令内のいずれかの小命令がNOP命令である場合に、そのNOP命令自体を削除する技術を開示している。すなわち、各VLIW命令の小命令毎に、NOP命令の数（以下、NOP数とも呼ぶ）を保持するフィールドを設け、その小命令が制御する機能ユニットにより、その小命令の直前に実行されるべきであったNOP命令の数をこのフィールドに格納し、その小

命令が属するVLIWに先行する一つ又は複数の連続するVLIWに含まれていた一つ又は複数のNOP命令を、それらの先行する一つ又は複数のVLIW命令から削除する。すなわち、先行するVLIW中の有効な小命令をその機能ユニットで実行した後に、このNOP数により決まるサイクル数だけ経過するまで、その小命令の実行開始を遅延する。この方法では、各命令フィールドごとに、先行するNOP命令を削除できるので、第1の参考文献の場合に比べて削減できるNOP命令の総数が多くなり、VLIW命令の総数が削減され、しかも、削除されたNOP命令に代えて削除されたNOP命令の数を保持するだけでよいので、各VLIW命令の長さはそれほど増大しない。従って、このようなVLIW命令により構成されたプログラムの容量は、この技術を採用しない場合に比べてかなり削減できる。

【0007】さらに、特開平7-105003号明細書（以下、第4の参考文献と呼ぶ）には、NOP命令以外の命令を含めて、VLIW命令列を圧縮し、圧縮されたVLIW命令列を主記憶等に記憶し、それらのVLIW命令列を展開した上で、実行する技術が開示されている。すなわち、プログラム中の、異なる構造のVLIW命令の各々に対応して、一つの変長の符号列を決定し、互いに同じ構造を有する複数のVLIW命令の各々をその構造に対して決定された符号列で置き換える。こうして得られた複数の符号列からなる、圧縮されたプログラムを主記憶に保持する。主記憶とは別に設けた命令デコード用のメモリには、圧縮されたプログラムに含まれる複数の符号列のいずれかに対応する圧縮されていない複数のVLIW命令を記憶し、上記圧縮されたプログラムを実行するときに、その圧縮されたプログラムを構成する複数の符号列の各々に基づいて、その符号列に対応する圧縮されていないVLIWを上記命令デコード用のメモリから読み出し、実行する。なお、上記符号列の決定に当たっては、複数のVLIW命令の構造が同じであるためには、それらの対応する小命令を構成する命令コードおよびオペランドの値も同じである必要がある。この技術では、各VLIW命令が、それより短い符号列に置き換えられるので、この技術はVLIW命令を空間的に圧縮する技術と言うこともできる。

【0008】

【発明が解決しようとする課題】上記第4の参考文献に記載の従来技術では、NOP命令以外の命令を含めてVLIW命令を圧縮するため、上記第1から第3の参考文献に記載の従来技術よりは、より小さなプログラムを得ることが出来るであろうと期待される。しかし、この第4の参考文献に記載の技術では、小命令中のレジスタ指定などを行うオペランドフィールドまで含めてVLIW命令の構造を判定しているため、プログラム中のVLIW命令の内、同一の構造と判定されるVLIW命令の数が多くはならない。そのため圧縮率はさほど大きくなら

ない恐れがある。

【0009】従って、本発明の目的は、より圧縮率の高い方法で圧縮されたVLIW命令列を実行するためのプロセッサを提供することである。

【0010】

【課題を解決するための手段】上記目的を達成するため、本発明によるVLIW命令用プロセッサで使用する圧縮されたVLIW命令は、複数の機能ユニット群の一つにそれぞれ対応した複数の命令フィールドを有し、各命令フィールドは、その命令フィールドに対応する機能

ユニット群により並列に実行されるべき一群の小命令を該一群の小命令よりも少ない情報でもって表す圧縮されたグループ命令を含む。

【0011】本発明によるVLIW命令用プロセッサは、複数の機能ユニットを含み、これらの機能ユニットは複数の機能ユニット群に区分される。上記プロセッサでは、上のVLIW命令を展開するための命令展開ユニットが、それぞれ該複数の機能ユニット群の一つに対応して設けられた、互いに並列に動作する複数の命令展開回路を有する。

【0012】各命令展開回路は、命令供給ユニットにより供給された一つの長命令内の、その各命令展開回路に対応する一つの機能ユニット群に対応する一つの命令フィールドに含まれた圧縮されたグループ命令を展開して、その対応する機能ユニットにより実行すべき一群の小命令を生成し、生成された一群の小命令を上記対応する一つの機能ユニット群に並列に出力する。

【0013】より具体的には、圧縮されたグループ命令は、そのグループ命令により表される一群の小命令の一つがそれぞれ必要とする一群のオペコードを、それら

の一群のオペコードよりも少ない情報で表す圧縮されたグループコードと、それぞれ該一群の小命令の一つが必要とする少なくとも一つのオペランドを表す一群のオペランド情報とを含む。

【0014】各命令展開回路は、該命令供給ユニットにより供給された上記一つの長命令内の、その命令展開回路に対応する命令フィールドに含まれた圧縮されたグループコードを展開してそのグループコードにより表される一群の小命令が必要とする一群のオペコードを生成して並列に出力するグループコード展開回路と、該出力された一群のオペコードの各々に上記一群のオペランド情報の一つを結合して一群の小命令を生成し、生成された一群の小命令を、その各命令展開回路に対応する上記一つの機能ユニット群に並列に出力する小命令生成ユニットとを有する。

【0015】さらに、本発明によるプロセッサのより具体的な態様では、各長命令の各命令フィールドは、そこに含まれた圧縮されたグループ命令内の圧縮されたグループコードが表す一群の小命令に対応する一群のNOP数を更に有する。各NOP数は、該一群の小命令内のそ

の各NOP数に対応する一つの小命令を実行した後に、その対応する小命令を実行する一つの機能ユニットにより実行すべきNOP命令の数を表す。

【0016】各命令展開回路内の上記小命令生成ユニットは、その各命令展開回路に対応する機能ユニット群内の複数の機能ユニットに対応して設けられた一群の小命令生成回路を有する。

【0017】各小命令生成回路は、小命令選択回路と制御回路とを有する。小命令選択回路は、上記長命令供給回路から供給された一つの長命令内の一つのグループ命令の圧縮されたグループコードに対してその各命令展開回路内の上記グループコード展開回路により生成された一群のオペコードの一つと、その一つのグループ命令内の一群のオペランド情報の一つとを結合して、一つの小命令を生成し、その各小命令生成回路に対応する一つの機能ユニットに供給する。さらにその供給の後に、上記一つのグループ命令に含まれる、一群のNOP数の一つにより指定される数のNOP命令を順次その小命令生成回路に対応する機能ユニットに供給する。制御回路は、上記小命令選択回路が上記各命令展開回路から上記一群のオペコードの後に供給される、上記一つの長命令の後続の長命令に対する後続の一群のオペコードの一つを含む小命令を上記対応する機能ユニットに供給するのを、上記一つのNOP数で指定される数のNOP命令の供給が完了するまで禁止する。

【0018】

【発明の実施の形態】以下、本発明に係わるVLIW命令用のプロセッサを図面に示したいくつかの実施の形態を参照して更に詳細に説明する。なお、以下においては、同じ参照番号は同じものもしくは類似のものを表すものとする。また、発明の第2の実施の形態以降については、発明の第1の実施の形態との相違点を主に説明する。

【0019】<発明の実施の形態1>

(1)装置の概略構成

図1において、VLIWプロセッサシステムは、主記憶100と、それに接続されたVLIWプロセッサ200からなる。このプロセッサ200には、同一のVLIW命令中の複数の小命令を並列に実行するための複数の機能ユニットの例として、たとえば、ロード用的小命令あるいはストア用的小命令を実行するためのロード/ストア(LD/ST)ユニット204-1、204-2、浮動小数点演算用的小命令を実行するための演算器205-1、2、固定小数点演算用的小命令を実行するための演算器206-1、206-2および分岐用的小命令を実行するための分岐ユニット207が設けられている。以下では、ロード用的小命令またはストア用的小命令を単にロード命令あるいはストア命令と呼び、更にこれらの命令を単にロード/ストア命令あるいはL/S命令と呼ぶことがある。同様に、浮動小数点演算用的小命令を単



に浮動小数点演算命令あるいはFL命令と呼ぶことがある。同様に、固定小数点演算用の小命令を単に固定小数点演算命令あるいはFX命令と呼ぶことがある。同様に、分岐用の小命令を単に分岐命令と呼ぶことがある。

【0020】主記憶100には、圧縮された複数の圧縮VLIW命令により構成されたプログラムが保持されている。命令フェッチユニット202は、命令キャッシュ201を介してこれらの命令を主記憶100から順次フェッチする。命令展開ユニット300は、フェッチされた各圧縮VLIW命令を展開して、複数の小命令からなる圧縮されていないVLIW命令を生成する本実施の形態に特徴的な回路である。本実施の形態では、複数のVLIW命令を時間的に圧縮することにより、これらの命令中のNOP命令を削減する。この結果得られるVLIW命令を時間的に圧縮されたVLIW命令と呼ぶ。更に、この時間的に圧縮されたVLIW命令の各々を空間的に圧縮する。この空間的な圧縮においては、時間的に圧縮された後のVLIW命令の各々を構成する複数の小命令を複数の群に分け、各群の複数の小命令を圧縮して新たに一つの圧縮されたグループ命令を生成する。こうして、各時間的に圧縮されたVLIW命令は、複数の圧縮されたグループ命令に変換される。以下では、この変換後のVLIW命令を空間的に圧縮されたVLIW命令と呼ぶことがある。

【0021】命令展開ユニット300は、各空間的に圧縮VLIW命令を展開して圧縮されていないVLIW命令を生成する回路で、具体的には、空間的に圧縮されたVLIW命令を構成する複数の圧縮されたグループ命令のいずれか一つをそれぞれ空間的及び時間的に展開して圧縮されていないVLIW命令を得るための複数の命令展開回路300a、300b、300cを有する。なお、分岐命令は、本実施の形態では、時間的に圧縮されるが、空間的に圧縮されない。命令展開回路300dは、そのような時間的に圧縮された分岐命令を時間的に展開する点で他の命令展開回路300a、300b、300cとは異なるが、以下の説明では、これらの命令展開回路300aから300dを区別しないで説明することもある。

【0022】デコードユニット203は、命令展開ユニット300から供給された圧縮されていないVLIW命令を構成する複数の小命令の各々を解説し、その小命令をいずれかの機能ユニット、すなわち、LD/STユニット204-1、204-2、浮動小数点演算器205-1、205-2、固定小数点演算器206-1、206-2、分岐ユニット207へ信号線405、406、407、408のいずれか一つに対して発行する、それ自体公知の回路である。

【0023】機能ユニット204-1、204-2、205-1、205-2、206-1、206-2、207の各々は、そこに対して発行された小命令が指定する

演算を、複数の汎用レジスタ（GR）と複数の浮動小数点レジスタ（FR）から成るレジスタ群（以下では、GR/FRレジスタ群と呼ぶ）208と内部バス416を使用して実行する。これらの機能ユニットの内、LD/STユニット204-1または204-2は、主記憶内のデータのロードを要求する命令である時あるいはその命令がデータを主記憶100に書き込むことを要求するストア命令が発行されたときに、内部バス416を介して、GR/FRレジスタ群208とデータキャッシュ208を制御して、それらのデータのロードあるいはストアを行う。浮動小数点演算器205-1または205-2または固定小数点演算器206-1または206-2は、浮動小数点演算命令がそこに発行されたときに、GR/FRレジスタ群208を使用してレジスタ演算を行う。また、分岐ユニット207は、分岐命令がそこに発行されたときに、その分岐命令が指定する分岐を実行する。また、その分岐が分岐予測による分岐方向と異なる際、信号線415を介して命令展開回路300a、300b、300c、300dへ分岐方向が異なることの通達を行う。

【0024】（2）VLIW命令の圧縮方法

図2（A）は、本実施の形態で使用する圧縮されていないVLIW命令100のフォーマットを示す。この命令は、図1に示した7つの機能ユニットの一つにて実行される小命令を保持するため7つの命令フィールド1、2、...、7からなる。各命令フィールドに保持される小命令は、有効な小命令もしくはNOP命令である。本実施の形態では、これらの命令フィールド1、2、...、7は、LD/STユニット204-1、204-2、浮動小数点演算器205-1、205-2、固定小数点演算器206-1、206-2、分岐ユニット207でそれぞれ実行される小命令に割り当てられる。有効な小命令は、操作の種類を示すオペレーションコード（以下では、簡単化のためにオペコードと呼ぶ）（OPC）用のフィールド10と、命令の操作が対象とする複数のオペランドOPD1、OPD2を保持するフィールド11a、11bを有している。

【0025】図2（B）は、図2（A）に示すフォーマットを有する複数のVLIW命令を時間的に圧縮して得られるVLIW命令100aのフォーマットを示す。この命令の各命令フィールド1a、2a、...、または7aには有効な小命令あるいはNOP命令が保持される。各小命令は、図2（A）に比べて、NOP数フィールド12を更に有している。この小命令がNOP命令であるときも同じである。各小命令のNOP数フィールド12には、その小命令を実行後に、その小命令が属するVLIW命令の後続のVLIW命令内の、その小命令が属する命令フィールドと同じ命令フィールドにある有効な小命令を実行する前に実行されるべきNOP命令の数を示す。すなわち、次の有効な小命令に対する遅延サイクル

数を示す。

【0026】すなわち、あい続く非圧縮のVLIW命令の同じ命令フィールドにNOP命令がある場合、それらのNOP命令の数を、これらのVLIW命令の直前のVLIW命令内の、そのフィールドに位置する有効な小命令に対するNOP数フィールドに記録する。言い換えると、異なるVLIW命令内に含まれた複数のNOP命令に代えて、この一つのNOP数を使用し、VLIW命令からこれらのNOP命令を除去する。従って、このようなNOP数を使用することにより、VLIW命令の数を削減できる。このため、本実施の形態では、このような処理を時間的な圧縮と呼んでいる。

【0027】図2(C)は、図2(B)に示すフォーマットを有する時間的に圧縮された一つのVLIW命令を空間的に圧縮して得られるVLIW命令100bのフォーマットを示す。この命令の各命令フィールド21、34、56または7aにはグループ命令が保持される。この空間的な圧縮においては、時間的に圧縮された一つのVLIW命令内の複数の小命令を、それぞれ複数の小命令からなる複数の群に分け、各群に含まれる複数の小命令を圧縮して、その小命令群を表す圧縮されたグループ命令を生成する。すなわち、各群のオペコードの代えて、それらのオペコードの組み合わせを表す、それらの組み合わせに一義的に対応するグループコードを使用する。このグループコードはこれらのオペコードの長さの総和より短くなるように選ぶ。従って、このグループコードは、これらの小命令のオペコードを圧縮したものになる。これらの小命令のオペランド情報はそのままグループ命令内に含まれる。

【0028】ところで、一つの小命令群に対して、一つの圧縮されたグループコードとそれらの小命令の複数のオペランド情報とを含む一つのグループ命令を生成する。従って、この新たなグループ命令で指定される複数の小命令の各々に付与したNOP数フィールドの長さを適当に短く選ぶと、このような新たなグループ命令を複数個持つVLIW命令は、時間的に圧縮されたVLIW命令より短くなる。従って、以下では、この新たなVLIW命令を空間的に圧縮した命令と呼ぶことがある。

【0029】具体的には、本実施の形態では、図2(B)に示す、ロード/ストア命令用の二つの命令フィールド1a、2a内の二つの小命令を空間的に圧縮して新たにグループ命令を生成する。グループ命令、たとえば、21は、小命令1a、2aのオペコードを圧縮して得られるグループコードを保持するフィールド10Aと、小命令1aのオペランド11a、11bを保持するオペランドフィールド11A、小命令1aのNOP数を保持するフィールド12A、小命令2aのオペランド11a、11bを保持するオペランドフィールド11B、小命令2aのNOP数を保持するフィールド12Bを有する。

【0030】同様に、図2(C)のフィールド21はこのようにして生成された新たなグループ命令を保持するフィールドである。同様に、浮動小数点演算命令用の二つのフィールド3a、4a内の二つの小命令を空間的に圧縮して新たなグループ命令を生成する。図2(C)のフィールド34はこのようにして生成された新たなグループ命令を保持するフィールドである。同様に、固定小数点演算命令用の二つのフィールド5a、6a内の二つの小命令を空間的に圧縮して新たなグループ命令を生成する。図2(C)のフィールド56はこのようにして生成された新たなグループ命令を保持するフィールドである。

【0031】本実施の形態では、後に詳細に説明するように、二つの小命令のオペコードに対する圧縮されたグループコードを生成するのに、これらの小命令フィールドのオペコードの組としてプログラム内に実際に出現するオペコードの異なる組を保持する命令テーブルを使用する。二つの時間的に圧縮されたVLIW命令内の特定の二つの小命令を空間的に圧縮するときに、それらの小命令のオペコードの組が保持されている、上記命令テーブル内の位置を表すポインタをそれらの小命令の二つのオペコードに代えてグループコードとして使用する。フィールド21、34、または56内のフィールド10Aにはこのグループコードを格納する。

【0032】なお、本実施の形態では、図2(C)に示された空間的に圧縮されたVLIW100c中には、図2(B)に示された分岐命令7aと一緒に空間的に圧縮するのに使用する他の小命令が存在しないので、この分岐命令7bは空間的に圧縮することなく、そのまま図2(C)に示された空間的に圧縮されたVLIW命令100c中に含まれる。

【0033】本実施の形態では、ソースプログラムから図1の圧縮されていないVLIW命令で記述された第1のプログラムを生成し、更に、この第1のプログラムを時間的に圧縮して、図2(B)に記載された時間的に圧縮されたVLIW命令で既述された第2のプログラムに変換し、更に、この第2のプログラムを空間的に圧縮して図2(C)に示す空間的に圧縮されたVLIW命令で既述された第3のプログラムを生成し、これを実行すべきプログラムとして使用する。以下、この圧縮の過程を具体的なプログラムを使用して説明する。

【0034】図8にベンチマークプログラムとして有名なリバモアカーネルカーネルループの1番を4倍展開したものを示す。このプログラムを図1に示す構造を有するプロセッサのためのアセンブラを用いてコーディングした結果、図9に示すプログラムが得られたと仮定する。図の左側に示す番号はこのプログラム中の各命令の番号である。

【0035】このプログラムで、第1番から第4番の命令LDは、それぞれロード命令であり、各々はその命令

10

20

30

40

50

の第1オペランドで指定されたレジスタ（ここでは第26、1、2、3番の浮動小数点レジスタFR26、FR1、FR2、またはFR3）へ、その命令の第2オペランドで指定された主記憶のアドレス（ここでは、配列2の(K+10)番目の要素が指し示すアドレス、または変数T、R、またはQで示されるアドレス）からデータをロードする命令である。

【0036】第5番の命令MOVEは、データ移動命令で、その第1オペランドのレジスタ（ここでは第4番の浮動小数点レジスタFR4）に第2オペランドのレジスタ（ここでは第26番の浮動小数点レジスタFR26）の内容をコピーする命令である。

【0037】第7、8、11、15、16、19、23、24、27、31、32、35番目の命令FMULTは、浮動小数点乗算命令で、その第2オペランドと第3オペランドで指定される二つの浮動小数点レジスタの内容の積をとり、その積を第1オペランドで指定される浮動小数点レジスタに格納する命令である。

【0038】第12、20、28、36番の命令FADDは、浮動小数点加算命令で、その第2オペランドと第3オペランドで指定される二つの浮動小数点レジスタの内容の和をとり、その和をその第1オペランドで指定される浮動小数点レジスタに格納する命令である。

【0039】第10、14、18、22、26、30、34番目の命令LDUIは、ロードアップ命令で、通常のロード命令と同様に、この命令の第2オペランドにて示されるアドレスのデータを主記憶から読み出し、この命令の第1オペランドにて示されるレジスタに転送するとともに、通常のロード命令と異なり、このデータのロードを実行後、第2オペランドのアドレスを保持するレジスタ（図中では略）の値を指定のアドレスだけ増加させる。

【0040】第38番の分岐命令は、第0番の汎用レジスタGR0に設定された繰り返し回数だけ、ラベルLOOPを付された命令（ここでは第5の命令）に分岐する命令である。なお、図10では、この汎用レジスタに繰り返し回数を設定するための命令は省略している。このように、図9のプログラムは、第1から第4の命令までを実行した後は、ラベルLOOPを付された第5の命令から最後の分岐命令までを繰り返し実行するプログラムである。

【0041】図9の命令列が指定する処理が、図1のプロセッサにおいて図2(A)で示したフォーマットを有するVLIW命令の列で実行され、しかもそれらの命令列の実行完了に必要な時間が短縮されるように、図9の命令列をスケジューリングすると、図10に示す命令列が得られる。ここで、図1に示したプロセッサにおける、ロード命令の実行後、ロードデータが使用可能になるまでのサイクル数（ロードレイテンシ）を6、乗算命令、加算命令等の演算命令の実行後に、演算結果が使用

可能になるまでのサイクル数（レイテンシ）を2と仮定した。

【0042】この図10において、各行が一つの圧縮されていないVLIW命令を表す。図において、上の方に示されたVLIWが先に実行される。各小命令に含まれる情報は、図2(A)に示されたとおりである。図9の命令列には、固定小数点演算が含まれていないので、図10では全てのVLIW命令の二つの固定小数点用のフィールドには全てNOP命令が埋められている。

【0043】なお、図10において、2重線より上の部分は、図9の第1から第4の命令が要求する処理を実行するVLIW命令で、一度だけ実行される。2重線より下のVLIW命令列は、図9の第5の命令以下の命令が要求するよりを実行するためのVLIW命令列で、これらは繰り返し実行される。この2重線の使用法は、以下の図11、13においても同じである。図10から分かるように、図9の第1から第4の命令は、図10の第1から第6のVLIW命令により実行され、図9の第5の命令以降の命令は、図10の第7のVLIW命令以降のVLIW命令により実行される。

【0044】図11は、図10に示したVLIW命令列を、時間的に圧縮して得られるVLIW命令列を示す。すなわち、図11のVLIW命令の各々は、図2(B)のフォーマットを有する。図10の各VLIW命令の各小命令に新たにNOP数フィールドを追加し、その小命令に続くNOP命令がある場合には、そのNOP命令の数をそこに格納し、それらの後続のNOP命令を削除することにより、図11のVLIW命令列が得られる。図11では、このNOP数フィールドは括弧とその中の数値で示されている。たとえば、第11図の左側の2列の第3から第6のNOP命令は削除され、それに代えて、第2行の二つのロード命令のNOP数フィールドに削除されたNOP命令の数4が記載されている。

【0045】ただし、本実施の形態では、NOP数フィールドを3ビットと仮定しているので、後続のNOP命令の数が0～7個の場合のみ、これらの後続のNOP命令の数をNOP数フィールドで指定可能であり、それらの後続のNOP命令を全て削除できる。しかし、もし、後続のNOP命令の数が8以上の場合には、その小命令のNOP数フィールドには、値7を格納し、後続のNOP命令の内、7個を削除する。削除されない残りのNOP命令に対しては、次のVLIW命令内に一つのNOP命令を割り当て、このNOP命令のNOP数フィールドに更に残りのNOP命令の内、最大7個のNOP命令の数を保持させる。これらの残りのNOP命令の数が7を超えているときには、以下同様に行う。

【0046】なお、後続のNOP命令の数が7以下の場合でも、それらのNOP命令が全て削除されないことも起きる。すなわち、図10において、第1のVLIW命令から第6のVLIW命令内の第1の浮動小数点演算用

の小命令は全てNOP命令であり、その総数は6である。第1のVLIW命令から第6のVLIW命令内の第2の浮動小数点演算用の小命令についても同じである。【0047】これらのNOP命令の総数が7以下であるが、これらのNOP命令が属するVLIW命令が要求する処理を実現するには、すでに説明したように、図11の第1、第2のVLIW命令が必要である。従って、図10の浮動小数点用の第1から第6のNOP命令は、図11の浮動小数点演算用の小命令として示したように、これらの二つのVLIW命令内に分散しておかれ、この例では、最初のNOP命令のNOP数フィールドには、図10の第1のVLIW命令から第6のVLIW命令内の第1の浮動小数点演算用の小命令として含まれている6つのNOP命令の残りのNOP命令の総数4が格納されている。

【0048】このように、異なるVLIW命令内の互に対応するフィールドに連続してNOP命令が含まれ、それらの総数が7以下である場合でも、それらのNOP命令が含まれるVLIW命令内の有効な小命令が要求する処理を実行するのに必要な時間的に圧縮されたVLIW命令の総数以下にはこれらのNOP命令を削減しない。従って、本実施の形態でも、時間的に圧縮されたVLIW命令列にはNOP命令が完全に存在しないわけではない。

【0049】図11に示された時間的に圧縮されたVLIW命令列の各々をさらに空間的に圧縮して図12に示す命令列40を得る。この空間的に圧縮では、図11の各VLIWを構成する複数の小命令を複数群に分け、各群の複数の小命令を空間的に圧縮することにより、時間的に圧縮されたVLIW命令を空間的に圧縮する。本実施の形態では、時間的に圧縮されたVLIWを二つずつの小命令、すなわち、一対のL/S命令、一対のFL命令、一対のFX命令からなる小命令群と、残りの分岐命令からなる一つの小命令群に分ける。更に、同一群に属する複数の小命令のそれぞれのオペコードを圧縮したコードとして、その群の小命令が取りうるオペコードの複数の組み合わせを保持する命令テーブル内の、それらの小命令オペコードの組み合わせと同じ組み合わせが保持されている特定の位置を示すポインタを使用する。

【0050】図12(A)、(B)、(C)は、それぞれL/S命令群、FL命令群、FX命令群に対する命令テーブル30A、30B、30Cを示す。これらの命令テーブル、たとえば、L/S命令群に対する命令テーブル30Aは、図11の空間的に圧縮された後の二つのL/S命令の列に含まれている異なるオペコードの組み合わせを抽出して命令テーブルを形成する。FL命令群とFX命令群に対しても同様である。図12(A)、

(B)、(C)から分かるように、LD/ST命令群に対する命令テーブル30Aは、わずかに4つの要素からなり、FL命令群に対する命令テーブル30Bは、5つ

の要素からなり、FX命令群に対する命令テーブル30Cの要素は、NOPとNOPの組み合わせのみからなる。各命令テーブルの各行の欄外に示した行番号を、その行に示されたオペコードの組み合わせに対するグループコードとして使用する。

【0051】図12(D)は、このようにして定められたグループコードを使用して、図11のVLIW命令から生成された空間的に圧縮されたVLIWを示す。この図の各L/S命令群の最左列にこの命令群に対して定めたグループコードを示す。FL命令群についても同様である。なお、第7番のL/S命令群は、NOP命令となっている。このように、空間的に圧縮されたVLIW命令にもNOP命令が含まれる理由は、図11に示された時間的に圧縮されたVLIW命令の場合と同じである。また、第10番以降のL/S命令群のグループコードは、FFとなっているが、このコードを有する小命令は、無効な命令で、この命令がフェッチされても実行されない。従って以下では、無効命令あるいは単にFF命令と呼ぶ。NOP命令は、実行され、1サイクル後に終了する命令であるので、この無効命令は、NOP命令とは異なる。

【0052】このような無効命令を使用する理由は以下の通りである。時間圧縮の結果、使用している演算器(命令フィールド)に偏りがあると、演算器の間で有効な命令があるものとないものが生じる。VLIW計算機では全ての演算器のための命令を一括してフェッチするため、この有効な命令だけをフェッチすることが出来ず、有効な命令のない演算器に関しては無効な命令をいれておく必要がある。この無効な命令としてNOP命令を使用すると、NOP命令には1サイクルの実行時間があるため、特に繰り返し処理などでは、演算器の間で実行がずれていくことになるため、通常のNOPではない無効命令というものを使用する。この無効命令の挿入方法は、以下の通りである。コンパイラによるコード生成時に、分岐命令の分岐先となりうる箇所、すなわち、ラベルの付加されている箇所毎にコードを分割し、分割されたコードの個まり(以下、ブロックと呼ぶ)毎に時間圧縮を行う。ブロック内で使用されている演算器に偏りがある場合、各命令フィールドについて最も多く、有効な命令を持つ命令フィールドと同じ数の命令となるように、他の命令フィールドに関して末尾に無効命令を追加する。

【0053】図12(D)において、全てのFX命令群に対するグループコードが全てFFになっている。これは、本実施の形態では有効な命令が全くないと仮定しているため、上記無効命令の挿入方法により無効命令が使用されたためである。また、図12の分岐命令フィールドには、時間的に圧縮されたVLIW命令に含まれていた分岐命令がそのまま保持される。すなわち、この命令は、時間的に圧縮されるが、空間的に圧縮はされない。

【0054】このように、図12(D)に示される空間的に圧縮されたVLIW命令列40と図12(A)から(C)に示される複数の命令テーブル30Aから30Cが、図9あるいは図10に示したプログラムを実行するために本実施の形態で使用するプログラムである。ただし、このプログラムを実行するに先だてこれらの命令テーブル30Aから30Cをそれぞれ主記憶100から命令展開回路300a、300b、300cに読み出してそこでのグループ命令の展開に使用する。このために、それぞれの命令テーブルをロードするための命令を

【0055】(3) VLIW命令の展開

命令列40と複数の命令テーブル30Aから30Cは主記憶100に保持される。命令フェッチユニット202は、命令キャッシュ201を介して主記憶100より、命令列40内の圧縮された命令を順次フェッチする。命令展開ユニット300は、フェッチされた命令をこれらの命令テーブル30Aから30Cを利用して展開し、デコードユニット203へ供給する。命令テーブル30A、30B、30Cは、本実施の形態では、後に説明する方法により、命令展開ユニット300により主記憶100からあらかじめフェッチされる。

【0056】命令展開ユニット300は、時間的、空間的に圧縮された各VLIW命令に含まれる複数のグループ命令の各々を、時間的にも空間的にも圧縮されていない複数の小命令に展開するもので、空間的に圧縮された各VLIW命令に含まれる複数のグループ命令をそれぞれ展開するための複数の命令展開回路からなる。本実施の形態では、LD/ST命令用の命令展開回路300a、浮動小数点命令(FP命令)用の命令展開回路300b、固定小数点命令(FX命令)用の命令展開回路300c、分岐命令用の命令展開回路300dとからなる。これらの内、命令展開回路300aから300cの各々が展開する命令は時間的にも空間的にも圧縮されているが、命令展開回路300dが展開する分岐命令は、時間的に圧縮されているが空間的には圧縮はされていない。したがって、命令展開回路300aから300cは、時間的にも空間的にも圧縮されている命令を展開する点で同じ構造を有する。命令展開回路300dは、時間的に圧縮されているが空間的には圧縮はされていない分岐命令を展開するので、命令展開回路300aから300cに比べて簡単な構造となっている。命令フェッチユニット202は、フェッチした命令が空間的に圧縮された命令であるときには、その命令内の複数のグループ命令を、命令展開回路300aから300dに線403を介して並列に分配する回路(図示せず)を有する。なお、本実施の形態では、分岐命令を分岐予測に基づいてフェッチするが、その予測のための回路および予測に基づく分岐先命令のフェッチを起動する回路等は、簡単化のために省略している。

【0057】図3を参照するに、命令展開回路300aは、命令フェッチユニット202から線403を介して供給される圧縮されたグループ命令の複数のフィールドを保持するための複数のキューを有する。すなわち、グループコードキュー304は、この圧縮されたグループ命令のフィールド10Aに含まれるグループコードを線403-1を介して受け取り、オペランドキュー305-1は、この圧縮されたグループ命令のフィールド11Aに含まれる複数のオペランドOPD1、OPD2を線403-2を介して受け取り、NOP数キュー306-1は、この圧縮されたグループ命令のフィールド12Aに含まれるNOP数を線403-4を介して受け取り、オペランドキュー305-2は、この圧縮されたグループ命令のフィールド11Bに含まれる複数のオペランドOPD1、OPD2を線403-3を介して受け取り、NOP数キュー306-2は、この圧縮されたグループ命令のフィールド12Bに含まれるNOP数を線403-5を介して受け取る。これらのキューは、ファーストインファーストアウトのキューで、それぞれ最先に受理した情報を最先に出力するように構成されている。なお、比較器308とFFレジスタ309は、命令フェッチユニット403から供給されたグループ命令が、無効命令(FP命令)であるときに、この小命令に含まれる情報をグループコードキュー304等が取り込むのを禁止するのに使用される。

【0058】命令展開回路300aには、さらに、グループコードキュー304から出力されるグループコードを、対応する一群のオペコードに展開するグループコード展開回路301を有する。命令展開回路300aには、さらに、このグループコード展開回路301から読み出された一群のオペコードを保持するためのオペコードキュー302-1、302-2と、これらのオペコードキューから出力される複数のオペコードに基づいて、複数の小命令を生成し、その命令展開回路に対応する複数の機能ユニットに出力する小命令生成ユニット33が設けられている。この小命令生成ユニット33は、オペコードキュー302-1、302-2のそれぞれ対応して設けられた一群の小命令生成回路303-1、303-2とを有する。小命令生成回路303-1は、オペコードキュー303-1から出力されるオペコードとオペランドキュー305-1から出力されるオペランドにて線50-1上に構成される小命令に対応する機能ユニットに出力するかあるいはNOP命令をその機能ユニットに出力するかを、NOP数キュー306-1から与えられるNOP数に応じて切り替える。すなわち、線50-1上の小命令を選択した後、その小命令に付随するNOP数に等しいサイクル数の期間だけNOP命令を出力する。これにより、この小命令に付随するNOP数で示されるNOP命令を生成することになる。こうして、この小命令生成回路は、グループコード展開回路301によ

る空間的展開により得られた小命令をさらに時間的に展開することにより、空間的にも時間的にも圧縮されていない複数の小命令を生成する。他の小命令生成回路303-2は、オペコードキュー302-2、オペランドキュー305-2、NOP数キュー306-2、NOP命令レジスタ604に同様に応答する。

【0059】以下、以上の装置の動作を更に詳細に説明する。グループコード展開回路301には、図4に示すごとく、命令テーブル30Aを保持するための命令テーブルメモリ307が設けられている。本実施の形態では、このメモリは、命令テーブル30Aの全体を保持するに必要な容量を有すると仮定する。他の命令展開回路についても同様とする。図12(D)のプログラムの実行の前に、図12(A)から(C)に示す命令テーブル30A、30B、30Cをそれぞれ命令展開回路300a、300b、300c内の命令テーブルメモリにロードするための命令が実行される。それぞれの命令は、時間的、空間的に圧縮された命令100b内の一つのグループ命令の代わりに、プリロード用の小命令を含む。図2(D)は、命令展開回路300aのためのプリロード用の小命令を含む長命令100cを示す。この命令では、プリロード用の小命令をLD/ST群用の命令フィールド21を含む。このプリロード用の小命令のオペランドフィールドは、ロードすべき主記憶内の領域の先頭アドレス（今の例では命令テーブル30Aの先頭アドレス）17と、ロードすべきエントリの数18を含む。このプリロード用の小命令のオペコード16の長さおよびオペランドフィールドの長さ、および命令長は、図2(C)に示した一つのグループ命令のものと同一である。従って、以下におけるプリロード用の小命令の処理の説明にあっては、この命令をグループ命令と見なし説明することがある。この長命令の他のフィールドには、NOP命令を含む。しかし、実施の態様によっては、これらの他のフィールドは他のグループ命令あるいは他の命令展開回路のためのプリロード用小命令を含んでいてもよい。

【0060】命令フェッチユニット202は、命令列40のフェッチに先だって、命令展開回路300a用のプリロード用の長命令100c（図2(D)）と、それと同様の、他の命令展開回路のためのプリロード用の小命令を含む長命令を順次フェッチする。たとえば、命令テーブル30Aのプリロードのための命令100cが命令フェッチユニット202によりフェッチされると、この命令の複数のグループ命令フィールドは、それぞれ命令展開ユニット300内の各命令展開回路300a、300b、300c、300dに転送される。

【0061】各命令展開回路、たとえば、300aのFFレジスタ309には、あらかじめ無効命令（FF命令）のグループコードFFが格納されている。命令フェッチユニット202から線403を介して転送されたブ

リロード用の小命令が供給されたときに、比較器308は、線403-1を介して供給される、この転送されたグループ命令のオペコードをこのレジスタ309内のFFと比較する。この比較の結果、もし一致が検出されなかったときには、この命令に含まれた情報の取り込みを、キュー304、305-1、306-1、305-2、306-2に許可する。すなわち、転送されたプリロード命令内の二つのオペランドは第1のオペランドキュー305-1、第1のNOPキュー306-1、第2のオペランドキュー305-2、第2のNOPキュー306-2に線403-2から403-5を介して分散して転送され、それらに分散して格納される。この命令のオペコードは線403-1を介してグループコードキュー510に転送され、そこに格納される。この時点より前では、これらのキューは空であると仮定すると、この命令のオペコードは直ちに線500を介してグループコード展開回路301aに転送され、また、二つのオペランドは、上に述べたいくつかのキューから線506を介してグループコード展開回路301に転送される。

【0062】図4を参照するに、グループコード展開回路301内のコマンドレジスタ808に、あらかじめプリロード命令のオペコードが格納されている。グループコードキュー304から線500を介してオペコードが転送されると、このオペコードがコマンドレジスタ808内のオペコードとが一致するか否を比較器807で検査することにより、転送されたオペコードがプリロード命令に対するものであるかを判断する。

【0063】転送されたオペコードとコマンドレジスタ808内のオペコードと一致した場合、比較器807の出力810が1になる。アドレスレジスタ801とエレメント数レジスタ802は、線506を介して供給された、プリフェッチすべき命令テーブルの先頭アドレスとその命令テーブル内の有効エントリ数とを取り込む。さらにメモリアクセス回路310は、比較器807の出力810により起動され、主記憶100内の、アドレスレジスタ801に保持されたアドレスを有する記憶位置とそれに続く記憶位置をエントリ数レジスタ802内のエントリ数だけ線402を介してアクセスして、アクセスされたこれらの記憶位置に保持された命令テーブル30Aを読み出し、線402を介してメモリ307に格納する。こうして命令テーブル30Aの読み出しが完了する。他の命令テーブル30B、30Cのプリロード用の命令を含む長命令もその後順次主記憶100からプリフェッチされ、それぞれのプリロード用の小命令が命令展開回路300b、300cにより読み出される。

【0064】その後、図12(D)に示すVLIW命令列が主記憶100から命令フェッチユニット202により順次フェッチされる。これらの命令の複数のグループ命令は、それぞれ命令展開回路300aから300dに転送される。各命令展開回路では、先に説明したプリロ

10

20

30

40

50

ード用の命令の場合と同様に、転送されたグループ命令が無効命令（FF命令）であるかを比較器308でチェックする。このグループ命令が無効命令（FF命令）でないときには、グループコードキュー304、オペランドキュー305-1、NOP数キュー306-1、オペランドキュー305-2、NOP数キュー306-2はそれぞれ線403-1から403-5より供給されるグループコード、グループ命令の生成に使用された第1の小命令のオペランド群、この第1の小命令のNOP数、上記グループ命令の生成に使用された第2の小命令のオペランド群、この第2の小命令のNOP数をそれぞれ取り込む。この際、各キューはキューの溢れを検出すると、信号線418-1に命令フェッチユニット202に対して後続の命令の送信の中断を要求する信号を出力する。この信号はORゲート518により他の命令展開回路300b、300c、300dからの同様の信号418-2、418-3、418-4とORされて、線418を介して命令フェッチユニット202に送られる。この結果、いずれの命令展開回路のいずれのキューも、命令フェッチユニット202から供給された小命令内の情報を取りこぼすことはない。

【0065】グループコード展開回路301には、線500によりグループコードキュー304の先頭にあるグループコードが入力される。このグループコード展開回路301では、命令テーブルメモリ307は、そこに保持された命令テーブル30A内の、この入力されたグループコードで指定される位置に保持された一群のオペコード内の複数のオペコードを出力する。比較器807の出力は、インバータ804を介してスイッチ806-1、806-2に供給され、比較器807の出力が0のときには、すなわち、線500を介して供給されたオペコードが上記プリロード命令に対するものでないときには、これらのスイッチは、命令テーブルメモリ307から読み出された複数のオペコードを線504-1、504-2を介してオペコードキュー302-1、302-2へ供給する。なお、比較器807の出力は、線505を介して命令選択回路303-1にも供給され、比較器807の出力が0のときには、すなわち、線500を介して供給されたオペコードが上記プリロード命令に対するものであるときには、1となり、グループコード展開回路301の出力は無効であることを小命令生成回路303-1に通知する。

【0066】オペコードキュー302-1、302-2は、ファーストインファーストアウトのキューで、それぞれに入力された複数のオペコードの内、最先に入力されたオペコードをそのキューの先頭の記憶位置に保持し、その先頭に位置するオペコードをそれぞれ線501-1、501-2に出力するように構成されている。オペコードキュー302-1から線501-1に出力されたオペコードと、オペランドキュー305-1から線5

02-1に出力された、オペランドキュー305-1内の先頭の記憶位置に保持されたオペランド群は、組み合わされて一つの小命令が復元される。この復元された小命令は小命令生成回路303-1に線501-1を介して供給される。

【0067】図5に示すように、小命令生成回路303-1は、線501-1から入力された小命令とレジスタ604から線605を介して入力されるNOP命令の一方を選択するためのセレクト600と、このセレクトの選択動作を制御する回路620を含む。この制御回路620内の、カウンタ601は、線503-1を介してNOP数キュー306-1（図3）から与えられるNOP数を取り込み、この数に等しい数のNOP命令を生成するためのものである。このカウンタ601の初期値は0にされる。比較器609は、このカウンタ601の値をレジスタ610内の定数0と比較し、一致が検出されたときに、値1を出力する。従って、小命令が線501-1に与えられた時点では、この比較器609の出力は1となる。この出力はANDゲート606を介してANDゲート608に供給される。ANDゲート606は、比較器609の出力とグループコード展開回路からの線505のANDをとることにより、線505が0である場合に、ANDゲート608の出力を0とさせてセレクト601にNOPを出力させるためのものである。ANDゲート608は、図1の装置の基本動作クロックである線611を介して与えられるクロックCLK1に同期して、比較器609の出力1をセレクト600に送る。セレクト600は、この信号に応じて線501-1を介して与えられる小命令を選択し、線404-1を介して対応する機能ユニットに送る。

【0068】ANDゲート614は、このANDゲート606の出力1と、信号線612で与えられる、半周期ずれたクロックCLK2に同期して、線503-1を介して与えられるNOP数の取り込みをカウンタ601に指示する。こうして、小命令が線501-1に与えられた時刻より半サイクル後に、カウンタ609は、このNOP数を取り込む。この結果、その後は比較器609の出力は0となる。従って、次にクロックCLK1が与えられた時点では、ANDゲート610の出力は0となり、セレクト600は、NOP命令レジスタ604内のNOP命令を選択し、線404に出力する。ANDゲート616にはこの比較器609の反転信号とクロックCLK1が入力されているので、このときのクロックCLK1にตอบสนองして、カウンタ601を1だけカウントダウンする。以下同様にして、NOP命令をこのカウンタ601の内容が0になるまで繰り返す。こうして、小命令の生成とその後の複数のNOP命令が生成される。なお、比較器609での比較が成立する毎に、その比較器の出力は、第1オペランドキュー305-1、オペコードキュー302-1に線508-1を介して転送され、さらに



グループコードキュー304に、ORゲート509を介して転送され、それぞれのキューから出力されている情報が選択されたことをこれらのキューに通知する。これらのキューは、この通知に回答して、現在出力している情報の次の情報を出力するように、出力する情報を切り替える。小命令生成回路303-2からの同様の出力も第1オペランドキュー305-2、オペコードキュー302-2に線508-2を介して転送され、さらにグループコードキュー304に、ORゲート509を介して転送される。こうして、一つの復元された小命令が小命令生成回路303-1で選択される毎に、以上に述べたキューは次のVLIW命令に属する情報を出力する。なお、命令展開回路300aに前述のプリロード命令が転送されたときには、すでに述べたように、小命令生成回路303-1に、線505を介してそのことが通知される。

【0069】以下では、図12(D)に示した空間的に圧縮された命令列の内、L/S命令群が図3の命令展開回路300aにより展開される様子を図13、図14、図15を参照して具体的に説明する。図13、図14、図15には、グループコードキュー304、オペランドキュー305-1、305-2、オペコードキュー302-1、302-2、NOP数キュー306-1、306-2の内容と、小命令生成回路303-1内のNOP数カウンタ601-1および小命令生成回路303-1内のNOP数カウンタ601-2の内容と、各サイクルで小命令生成回路303-1、303-2からデコードユニット203に出力される小命令が示される。

【0070】まず第1サイクルに、命令フェッチユニット202から、図12(D)の命令(1)が命令展開ユニット300に転送される。この命令内のL/S命令を命令展開回路300aが受け取ると、この命令のグループコード"1"、第1のオペランド群"FR26、2(K+10)"、第1のNOP数"0"、第2のオペランド群"FR1、T"、第2のNOP数"0"がグループコードキュー304、第1オペランドキュー305-1、第1NOP数キュー306-1、第2オペランドキュー305-2、第2NOP数キュー306-2へそれぞれ格納される。これらのキューはそれぞれに格納された情報が最初の情報であるので、それらの情報をそのまま出力する。グループコード展開回路301は、グループコードキュー304から供給されたグループコードが"1"であることから、命令テーブル回路内の命令テーブル30Aの第1エントリにある一対のオペコード"LD、LD"をオペコードキュー302-1、302-2へ出力する。第1、第2のオペコードキュー302-1、302-2はそれぞれに供給されたオペコード"LD"、"LD"がそれぞれに供給された最初のオペコードであるので、それぞれのオペコードをそれぞれ線501-1、501-2に出力する。

【0071】小命令生成回路303-1は、第1オペコードキュー302-1から供給されたオペコードと第1オペランドキュー305-1から出力されたオペランドにより構成される復元された小命令"LD FR26、2(K+10)"を選択し、デコードユニット203へ出力する。さらに、第1のNOP数キュー306-1から供給される第1のNOP数"0"をNOP数カウンタ601-1に取り込む。小命令生成回路303-1が復元された小命令を選択すると、小命令生成回路303-1に接続されたいろいろのキュー304、305-1等は、それぞれ次のVLIW命令のための情報を出力するように出力する情報を切り替える。小命令生成回路303-2も同様に小命令"LD FR1、T"を出力し、その中のNOP数カウンタ601-2に、第2のNOP数キュー306-2から供給される第2のNOP数"0"を取り込む。

【0072】第2サイクルには、図12の命令(2)が命令フェッチユニット202より命令展開ユニット300に送られ、第1サイクルと同様に処理される。この命令(2)は命令(1)と異なり、第1、第2のNOP数は"4"である。従って、命令展開回路300aでは、この命令の展開後には、NOPカウンタ601-1、601-2には"4"が格納される。

【0073】第3サイクルでは、図12の命令(3)が命令フェッチユニット202より命令展開ユニット300に送られ、第1サイクルと同様に各キューにこの命令の情報が格納される。しかし、命令展開回路300aでは、このサイクルでは、NOPカウンタ601-1、601-2の値が0ではないために、小命令生成回路303-1、303-2はともにレジスタ604から与えられるNOP命令を選択し、デコードユニット203へ出力する。更に、NOPカウンタ601-1、601-2の値をデクリメントする。小命令生成回路303-1、303-2は命令フェッチユニット202より新たに転送された命令(3)を選択しない。従って、このサイクルでは、命令(3)は展開されないの、各キューに保持されたこの命令の情報は、各キューにそのまま保持される。

【0074】第4~6サイクルでは、第3サイクルと同様に、図12の命令(4)~(6)が順次命令フェッチユニット202から命令展開ユニット300に供給される。命令展開回路300aでは、これらのサイクルでは、第3サイクルと同様に、NOPカウンタ601-1、601-2の値が0ではないために、小命令生成回路303-1、303-2はともにレジスタ604から与えられるNOP命令を選択し、更に、NOPカウンタ601-1、601-2の値をデクリメントする。各キューに保持された命令(4)~(6)に関連する情報は、そのままそのキューに保持される。第6サイクルの終了時にNOPカウンタ601-1、601-2の値が



0となる。こうして、図10に示す、最初の6つのVLIW命令に属する6対のLD/ST命令が復元される。  
 【0075】次に第7サイクルでは、図12の命令(7)が命令フェッチユニット202より命令展開ユニット300に送られる。この命令(7)の内、LD/ST用の小命令は一对のNOP命令を空間的に圧縮した命令であるので、オペランドを取らないため、図12では空欄で示しているが、実際にはNOP命令のパターンでオペコードフィールドを除いたものが入っている。

【0076】命令展開回路300aでは、第7サイクルではNOPカウンタ601-1、601-2の値がすでに0であるため、第1サイクルと同様に各キューの先頭に保持された情報に基づいて圧縮されていない命令が復元される。今の例では、命令(3)に関する情報に基づいて、この命令に対応する、図10の第7のVLIW命令を構成する一对のLD/ST命令が復元される。第8サイクルでも、同様に図12の命令(8)が命令フェッチユニット202より命令展開ユニット300に送られ、命令展開回路300aでは、第7サイクルと同様に処理される。すなわち、この第8サイクルでは、命令(4)に関する情報に基づいて、この命令に対応する、図10の第8のVLIW命令を構成する一对のLD/ST命令が復元される。この復元後に命令(8)の第1、第2のNOP数"2"、"2"がNOPレジスタ601-1、601-2に格納される。

【0077】第9、10サイクルでは、図12の命令(9)、(10)が命令フェッチユニット202より命令展開ユニット300に送られる。命令展開回路300aでは、これらのサイクルではNOPカウンタ601-1、601-2の値が0でないため、第3サイクル等と同じく、小命令生成回路303-1、303-2によりNOP命令が出力される。NOPカウンタ601-1、601-2の値は第10サイクルの間に0になる。

【0078】なお、第10サイクルに命令フェッチユニット202より転送される命令(10)は、無効命令(FF命令)である。比較器308が命令(10)のオペコードポインタセットがFFレジスタ309内のFFと同じであることを検出すると、命令展開回路300a内の各キューはこの命令に関する情報を取り込まない。このような無効命令の扱いは、第11から第13サイクルでも同じである。

【0079】第11サイクルでは、命令展開回路300aでは、NOPカウンタ601-1、601-2の値が0であるため、第7サイクルと同様に、各キューの先頭に保持された情報に基づいて圧縮されていない命令が復元される。今の例では、命令(5)に関する情報に基づいて、この命令に対応する、図10の第11のVLIW命令を構成する一对のLD/ST命令が復元される。この復元後、NOPカウンタ601-1、601-2には命令(5)の第1、第2のNOP数"0"、"0"がセ

ットされる。したがって、次の第12サイクルでも、命令展開回路300aでは、NOPカウンタ601-1、601-2の値が0であるため、第11サイクルと同様に、各キューの先頭にある情報の基づいて、命令、今の例では命令(6)に対応する、図10の第12のVLIW命令を構成する一对のLD/ST命令が復元される。この復元後、NOPカウンタ601-1、601-2には命令(6)の第1、第2のNOP数"7"、"7"がセットされる。

【0080】次の第13サイクルから第19サイクルの間は、命令展開回路300aでは、NOPカウンタ601-1、601-2の値が0でないため、第3から第6サイクルと同様にNOP命令が繰り返し出力され、第19サイクルでのNOP命令の出力後に、NOPカウンタ601-1、601-2の値が0となる。

【0081】なお、第14サイクル以降では、フェッチステージからは分岐予測先が確定していれば、その命令を入力として順次命令展開回路300aから300dへ渡される。分岐先が確定していない場合には入力を無しとする。ただし、分岐予測先が確定しており、各命令展開回路の各キューに予測された命令が入っている場合には、予測された分岐の方向と異なる方向の分岐が生じると、各命令展開回路へ予測ミスであることが分岐ユニット207から信号線415を介して通知され、各命令展開回路では各キュー内で命令の無効化を行う。

【0082】第20サイクルでは、命令展開回路300aは、NOPカウンタ601-1、601-2の値が0であるため、各キューの先頭にある情報の基づいて、命令、今の例では命令(7)(これは一对のNOP命令を空間的に圧縮した命令である)に対応する、図10の第20のVLIW命令を構成する一对のNOP命令が復元される。この復元後、NOPカウンタ601-1、601-2にはこの命令(7)の第1、第2のNOP数"0"、"3"がセットされる。従って、NOPカウンタ601-1、601-2の値が異なる結果、以後は、小命令生成回路303-1、303-2の動作は異なる。

【0083】第21サイクルでは、命令展開回路300aでは、NOPカウンタ601-1の値が0であるため、小命令生成回路303-1は、このユニットに対応するキュー、すなわち、オペコードキュー302-1、第1オペランドキュー305-1の先頭に保持された情報、今の例では、命令(8)を生成するのに使用された第1の小命令に関する情報から、この第1の小命令"STU FR11, X(K)"を復元し、その復元後、第1のNOP数キュー306-1の先頭位置に保持された第1のNOP数、今の例では"0"をNOPカウンタ601-1にセットする。一方、小命令生成回路303-2は、NOPカウンタ601-2の値が0でないため、NOP命令を出力し、このNOPカウンタをデクリメントする。この結果、命令展開回路300aからは、有効

な小命令とNOP命令の対からなる一部のLD/ST命令群が復元される。この命令群は、第11図の第21の命令を構成する命令群である。このように、本実施の形態では、空間的に圧縮された命令の一部を利用して、有効な小命令とNOP命令を含む小命令群も復元できる。この空間的に圧縮された小命令の残りの部分は、以下に説明するように、他の有効な小命令とNOP命令を含む他の小命令群の復元に利用できる。

【0084】第22サイクルでは、命令展開回路300aでは、NOPカウンタ601-1の値が0であるため、小命令生成回路303-1は、第21サイクルと同様に動作し、オペコードキュー302-1、第1オペランドキュー305-1の先頭に保持された命令(9)を生成するのに使用された第1の小命令に関する情報から、この第1の小命令“STU FR18, X(K+1)”を復元し、その復元後、第1のNOP数キュー306-1の先頭位置に保持された第1のNOP数、今の例では“3”をNOPカウンタ601-1にセットする。一方、小命令生成回路303-2は、NOPカウンタ601-2の値が0でないため、NOP命令を出力し、このNOPカウンタをデクリメントする。

【0085】第23サイクルでは、命令展開回路300aでは、NOPカウンタ601-1、601-2の値がともに0でないため、小命令生成回路303-1、303-2はNOP命令を出力し、それぞれNOPカウンタ601-1、601-2をデクリメントする。

【0086】第24サイクルでは、命令展開回路300aでは、NOPカウンタ601-2の値が0であるため、小命令生成回路303-2は、このユニットに対応するキュー、すなわち、第2オペコードキュー302-2、第2オペランドキュー305-2の先頭記憶位置に保持された情報、今の例では、命令(8)を生成するのに使用された第2の小命令に関する情報から、この第2の小命令“STU FR25, X(K+2)”を復元し、その復元後、第2のNOP数キュー306-2の先頭位置に保持された第2のNOP数、今の例では“0”をNOPカウンタ601-2にセットする。小命令生成回路303-1は、NOPカウンタ601-1の値が0でないため、第23サイクルと同様にNOP命令を出力し、その後このカウンタをデクリメントする。この結果、命令展開回路300aからは、NOP命令と有効な小命令との対からなる一対のLD/ST命令が復元される。この命令群は、第11図の第24の命令を構成する命令群である。このように、本実施の形態では、先に第21サイクルで部分的に復元された命令(8)の残りの情報を利用して、新たに有効な小命令とNOP命令との対を復元できる。

【0087】第25サイクルでは、命令展開回路300aでは、第24サイクルと同様にして、小命令生成回路303-1、303-2は、それぞれNOP命令と小

命令“STU FR32, X(K+2)”を復元する。ここで、25サイクル目では分岐命令が実行されているため、分岐先が決定し、分岐予測が正しいか否かが判定できるので、分岐予測が正しければキューに入っているものを使用して上記の手順により、命令展開を行う。分岐予測が正しくない場合、キューに入っている命令で分岐予測に基づいているものを無効化し、図13の1サイクル目からと同様の処理により命令展開を行うことが可能となる。以上から分かるように、本実施の形態では、時間的、空間的に圧縮されたグループ命令を展開して複数の圧縮されていない小命令および複数のNOP命令を復元できる。

【0088】ここで、図2(A)の圧縮されていないVLIWのオペコードフィールド10を12ビット、オペランドフィールド11を20ビットと仮定すると、各小命令の長さは32ビットである。したがって、図1(A)命令長が32ビット×7フィールド=224ビットであり、このVLIW命令で記述された図10のプログラム全体の容量は224ビット×25命令=700バイトとなる。図10のプログラムを図1(B)に示すVLIWを使用するように時間的に圧縮すると、図11に示すように必要な命令数は13に減少される。図11のプログラムを図1(C)に示すVLIWを使用するように空間的に圧縮すると、このプログラムの命令が取りうるオペコードの組み合わせは、L/S命令グループ、FL命令グループ、FX命令グループでは図12(A)(B)(C)に示すように、それぞれ4、5、1通りである。したがって、図1(C)空間的に圧縮されたVLIW命令のグループコード13は、3ビットで十分である。このVLIW命令の分岐命令フィールド7aにはグループコードは不要である。NOP数フィールド12を3ビットと仮定すると、このVLIW命令は、LD/ST命令、浮動小数点演算命令、固定小数点命令をそれぞれ2つと、分岐命令を1つもつので、VLIW命令の命令長は、3ビット(長)×3グループ+20ビット(オペランド長)×6フィールド(分岐命令以外)+3ビット(NOP数フィールド長)×6フィールド(分岐命令以外)+32(分岐命令)+3ビット(分岐命令用NOPフィールド)=182ビットとなる。

【0089】空間的に圧縮された後のプログラムは13命令からなるので、プログラム自体の大きさは約296バイトとなる。命令テーブル群30A、30B、30Cのエントリの合計は、10エントリであるので、これらのテーブル群の容量は30バイトとなる。プログラムと命令テーブル群の両者を併せても、元のプログラムに対して約53.4%の減少となる。しかも、命令テーブル群の容量は約30バイトと非常に小さく、第4の参考文献のもの(22通りのVLIW命令×32ビットの小命令×7フィールド=616バイト)と比較すると、21分の1になっているので、プロセッサ内のより高速なメ

メモリに保持することが可能となる。

【0090】この図10のコードについて、従来の技術で挙げた第4の参考文献による命令圧縮方法では、NOP数フィールドを持たないため小命令は32ビット長となり、22通りの異なる命令の組が存在し、共通化できるのは全てのフィールドがNOP命令のものだけとなっている。したがって、命令を表す符号として5ビットを用いると、プログラム全体としては、5ビット×25命令+32ビット×7フィールド×22通り=622バイト（内、命令テーブルは616バイト）となる。これは、元の大きさに比べて、約11%（文字化け）%の減少に過ぎない。また、第1の参考文献の方法については、VLIW命令毎に付加する遅延サイクル数フィールドを3ビットと仮定すると、この例では、全フィールドがNOPであるのは4命令なので、全体としては21命令に減少するが、1VLIW命令が32ビット×7フィールド+3ビット=227ビットになる。したがって、プログラム全体としては、約596バイトになり、元の大きさに比べて、約15%の減少となる。さらに、第2の参考文献の方法によれば、各小命令に付加する遅延サイクル数フィールドを第1の参考文献と同様に3ビットと仮定すると、1VLIW命令が32ビット×7+3ビット×7=245ビットとなる。一方、NOP削除の効果により、図11のようにプログラム全体が13命令となるから、プログラム全体では約398バイトとなる。これは、元の大きさに比べて、約43%の減少となっている。

【0091】＜発明の実施の形態2＞本実施の形態では、実施の形態1で使用したグループコード展開回路301内の、命令テーブルを保持するためのメモリ307の必要容量を低減する。図12（D）に示したプログラムの場合、命令テーブル30A、30B、30Cのサイズは異なり、命令テーブル30Aが最も大きい。命令テーブルの大きさは、使用されるプログラム内のオペコードの種類が多いほど大きくなる。実施の形態1では各命令展開回路の命令テーブルメモリは、一つの命令テーブル全体を持たなければならない。従って、大きな命令テーブルを保持すべき命令展開回路では、巨大な命令テーブルメモリを用意しなければならなくなる。そこで、本実施の形態では、実施の形態1に比べて小さな命令テーブルメモリを使用することを可能にする。すなわち、い

ずれかの命令展開回路内に格納すべき命令テーブルを複数の部分命令テーブルに分け、プログラムからの要求に応じて必要な部分命令テーブルをその命令展開回路内の命令テーブルメモリへ主記憶からコピーする。

【0092】たとえば、命令テーブル30Aを、図12

命令テーブルの先頭のをあらかじめ主記憶100からロードし、図12に示す空間的に圧縮されたプログラムの実行の過程で、適宜後続の部分命令テーブル部分を主記憶100からロードして使用する。このために、コンパイル時に、図12（D）のプログラムを複数のプログラム部分に分け、各部分プログラムに含まれるVLIW命令の一对のオペコード含む部分命令テーブルを生成し、主記憶100内にこのように作成した複数の部分命令テーブルを記憶する。さらに、各部分プログラムの先頭に、その部分プログラムで使用する部分命令テーブルを主記憶100からプリロードする命令を含む圧縮された長命令を埋め込む。この命令は、図2（D）に示した命令100dと同じフォーマットを有する。この実施の形態では、この命令の他のフィールドには、他の命令展開回路のための、NOP命令以外の有効なグループ命令を含めることが出来る。プロセッサの回路構成は実施の形態1のものと同じである。なお、本実施の形態による命令テーブルの構成は、全ての命令展開回路に適用する必要はなく、比較的大きな命令テーブルを使用する可能性のある命令展開回路のみに適用することでもよい。

【0093】＜発明の実施の形態3＞本実施の形態は、実施の形態2と同じく、命令テーブルメモリの容量を削減するが、実施の形態2と異なり、キャッシュタイプの命令テーブルメモリを使用する。

【0094】本実施の形態では、グループコード展開回路300aとして、図4のものを代えて図6のものを使用する。本実施の形態でも、図12（D）の命令列を主記憶100からフェッチする前に、命令展開回路300aに命令テーブル30Aに関連して図2（D）に示すプリロード命令を含む長命令を使用する。但し、本実施の形態でのプリロード命令は、エントリ数フィールド18には有効な情報を有しない。また、このプリロード命令は、命令テーブル30Aをロードするのではなく、ロード開始アドレス17をグループコード展開回路300a内にセットするのに使用される。

【0095】図6において、コマンドレジスタ808には、実施の形態1と同じく上記プリロード命令のオペコードがあらかじめ格納される。グループコードキュー304から線500を介してグループ命令が供給されると、そのオペコードとコマンドレジスタ808内のものとを比較器807で比較する。この比較の結果、一致が検出された場合、供給されたグループ命令を上記プリロード命令とみなし、このときに信号線506を介して供給されるロード開始アドレス17を命令テーブル30Aが格納されている領域のベースアドレスとしてベースレジスタ702に格納する。さらに、このときの比較器807の出力は、線810、ORゲート711、線505を介して小命令生成回路303-1、303-2に送られ、これらの回路にNOP命令の生成を指示するのは実施の形態1の場合と同じである。他の命令展開回路に

関しても同じようにプリロード命令が実行される。

【0096】その後、図12(D)に示す命令列が順次フェッチされる。これらの命令内のグループコードが線500を介して供給されるごとに、グループコードによりタグメモリ701と命令テーブルメモリ700を参照する。タグメモリ701は複数のグループコードをそれぞれ

のグループコードに対応する記憶位置に保持するためのもので、命令テーブルメモリ700は、タグメモリ701に登録された複数のグループコードに対応する、オペコード対を、それぞれのグループコードに記憶位置に保持するためのメモリである。

【0097】線500上のグループコードによりタグメモリ701を参照した結果、このグループコードがヒットしない時には、タグメモリ701はミスヒット信号を線709に出力するとともに、このグループコードをタグメモリ701内のこのグループコードに対応する位置に保持する。メモリアクセス回路703は、このミスヒット信号にตอบสนองして、ベースレジスタ702内のベースアドレスと線500上のグループオペコードとを加算して、主記憶アドレスを生成する回路(図示せず)を有し、さらに、このアドレスを使用して、命令テーブル30A内の、このグループコードに対応するオペコード対を線402を介して主記憶100より読み出す回路(図示せず)を有する。命令テーブルメモリ700は、ミスヒット信号にตอบสนองして、この読み出し回路により主記憶100から線402に読み出されたオペコード対を、線500上のオペコードに対応する記憶位置に保持し、さらに線504-1、504-2を介して非圧縮オペコードキュー302-1、302-2(図3)へ有効なオペコード対として供給する。なお、ミスヒット信号は線709、ORゲート711、線505を介して小命令生成回路303-1、303-2(図3)に送られ、これらの回路にNOP命令の生成を指示する。

【0098】このような動作が、グループコードキュー304より与えられる後続の複数のグループコードの各々に対して実行される。もし、線500から与えられたとき後続のグループコードによりタグメモリ701を参照した結果、このグループコードがヒットした時には、命令テーブルメモリ700は、そのグループコードに対応した記憶位置に保持されたオペコード対を線504-1、504-2を介して非圧縮オペコードキュー302-1、302-2(図3)へ有効なオペコード対として供給する。

【0099】このように、本実施の形態では、グループコードがタグメモリ701に登録されていないときのみ、その変換に使用するオペコード対を主記憶からフェッチするので、命令テーブルメモリ700の容量は実施の形態1、2より小さくすることが出来る。

【0100】<発明の実施の形態4>本実施の形態は、実施の形態2、3と同じく、命令テーブルメモリの容量

を削減するが、実施の形態2、3と異なり、プログラムでのオペコード対の使用頻度がオペコード対により異なることを利用する。すなわち、複数の頻出するオペコード対を固定的に保持する固定命令テーブルメモリと、動的に変化する複数の頻出しなないオペコード対をする可変命令テーブルメモリとを使用する。

【0101】図7において、901は固定命令テーブルメモリであり、900は可変命令テーブルメモリである。ここでは、固定命令テーブルメモリ901には予め定められた複数のグループコードに対応する複数のオペコード対を保持する。従って、このメモリ901には主記憶100から固定の命令テーブルをロードしない。しかし、本実施の形態では、実施の形態2と同様に、プログラムの実行にあわせて必要となる部分命令テーブルを可変命令テーブルメモリ900に格納するようにする。予め実行すべきプログラムを複数の部分プログラムに分け、それぞれに対応して部分命令テーブルを決定し、主記憶100に記録すること、それぞれの部分プログラムの先頭位置にそれぞれの部分プログラムで使用する部分命令テーブルをロードする命令を含む、図2(D)に示した長命令を含めることは実施の形態2と同じである。

【0102】しかし、本実施の形態では、各部分プログラムに対する部分命令テーブルとして、その部分プログラムで使用する一群のグループコードの内、命令固定命令テーブルメモリ901に格納された命令テーブルに含まれていない複数のグループコードに対応する複数のオペコード対を含む部分命令テーブルを使用する。グループコードについては、固定命令テーブルと可変命令テーブルの双方を表せられるビット数を持ち、固定命令テーブルに格納されるオペコード対に関しては、その最上位ビットが0であるようなグループコードにより表され、可変命令テーブルに格納されるオペコード対に関しては、その最上位ビットが1であるようなグループコードにより表されている。

【0103】各部分プログラムの先頭に設けられたプリロード命令により、部分命令テーブルが主記憶100より読み出されるのは、実施の形態2と同じである。本実施の形態4ではこの読み出された部分テーブルは、可変命令テーブルメモリ900に格納される。線500より供給されたグループコードにより二つの命令テーブルメモリ900、901が同時に参照され、読み出されたオペコード対のうち、適切なものをグループコードの最上位ビットによりセレクタ902により選択し、線504-1、504-2に出力され、さらに、非圧縮オペコードキュー302-1、302-2(図3)へ有効なオペコード対として供給する。以上から分かるように、この実施の形態では、部分命令テーブルのサイズを小さくでき、それでいて主記憶100から部分命令テーブルをロードする回数を少なくできる。

【0104】なお、本実施の形態において、プログラム

の実行前に固定の命令テーブルを主記憶100からメモリ901にロードするように変形することも可能である。また、メモリ900、901を実施の形態3で利用したキャッシュタイプのメモリにて構成することも可能である。

【0105】以上のいくつかの実施の形態に示したように、VLIW命令中の小命令のオペコードとオペランドフィールドに分け、オペランドフィールドに後続のNOP命令の数を記録するフィールドにその数を記録することにより、NOP命令の削減が行え、有効命令の密度を高め、主記憶と命令キャッシュの使用効率の向上を図ることが出来る。さらに、VLIW命令中の小命令中のオペコードフィールドの組み合わせの数を減らせることができ、これにより、同じオペコードフィールドの組み合わせを同一の符号により表すことが出来る確率が高くなる。このオペコードフィールドの共有により、命令並列度が高くなっても、VLIW命令としては、オペランドフィールドのみの増加となるため、従来の方法に比べて、VLIW命令長の増加を抑えることが出来る。

【0106】さらに、圧縮された命令の展開機構自身の構成も非常に簡略になっているため、マシンサイクルに悪影響を及ぼすようなことはないため、圧縮率の高い長命令を高速に実行できる。

【0107】さらに、NOP命令の削除による時間方向の圧縮により、有効な小命令は実際に実行されるよりも前にフェッチされているため、分岐命令やL/S命令に関しては、オペコードを前もって調べることができ、分岐先アドレスの予測やデータのプリフェッチが可能になるという効果がある。

【0108】

【発明の効果】本発明では、より圧縮率の高い方法で圧縮されたVLIW命令を実行するプロセッサが得られる。

【図面の簡単な説明】

【図1】本発明の第1の実施の形態によるVLIW命令用プロセッサの概略構成図。

【図2】(A)は圧縮されていないVLIW命令のフォーマットを示す図。(B)は時間的に圧縮されたVLIW\*

\*W命令のフォーマットを示す図。(C)は時間的にも空間的にも圧縮されたVLIW命令のフォーマットを示す図。(D)は命令テーブルのプリロード用の命令を含むVLIW命令のフォーマットを示す図。

【図3】図1の装置に使用する命令展開回路の概略回路図。

【図4】図3の装置に使用するグループコード展開回路の概略回路図。

【図5】図4の装置に使用する小命令生成回路の概略回路図。

【図6】本発明の他の実施の形態によるVLIW命令用プロセッサに使用するグループコード展開回路の概略回路図。

【図7】本発明のさらに他の実施の形態によるVLIW命令用プロセッサに使用するグループコード展開回路の概略回路図。

【図8】図1の装置で実行される処理を示すフォートランプログラムの例を示す図。

【図9】図8のフォートランプログラムを仮想プロセッサ用のアセンブリ言語を使って表現したプログラムの例を示す図。

【図10】図9のアセンブリ言語プログラムをスケジューリングして得られるプログラムの例を示す図。

【図11】図10の命令列で表される処理を実行する、図1(A)のフォーマットを有する長命令列により表現したプログラムを示す図。

【図12】(A)は、ロードストア命令用のテーブルの例を示す図。(B)は、浮動小数点演算命令用の命令テーブルの例Wを示す図。(C)は、固定小数点演算命令用の命令テーブルの例Wを示す図。(D)は、図11のプログラムを図1(C)のフォーマットを有する長命令により表現したプログラムの例を示す図

【図13】図12のプログラムの一部の実行の様子を説明する図。

【図14】図12のプログラムの他の一部の実行の様子を説明する図。

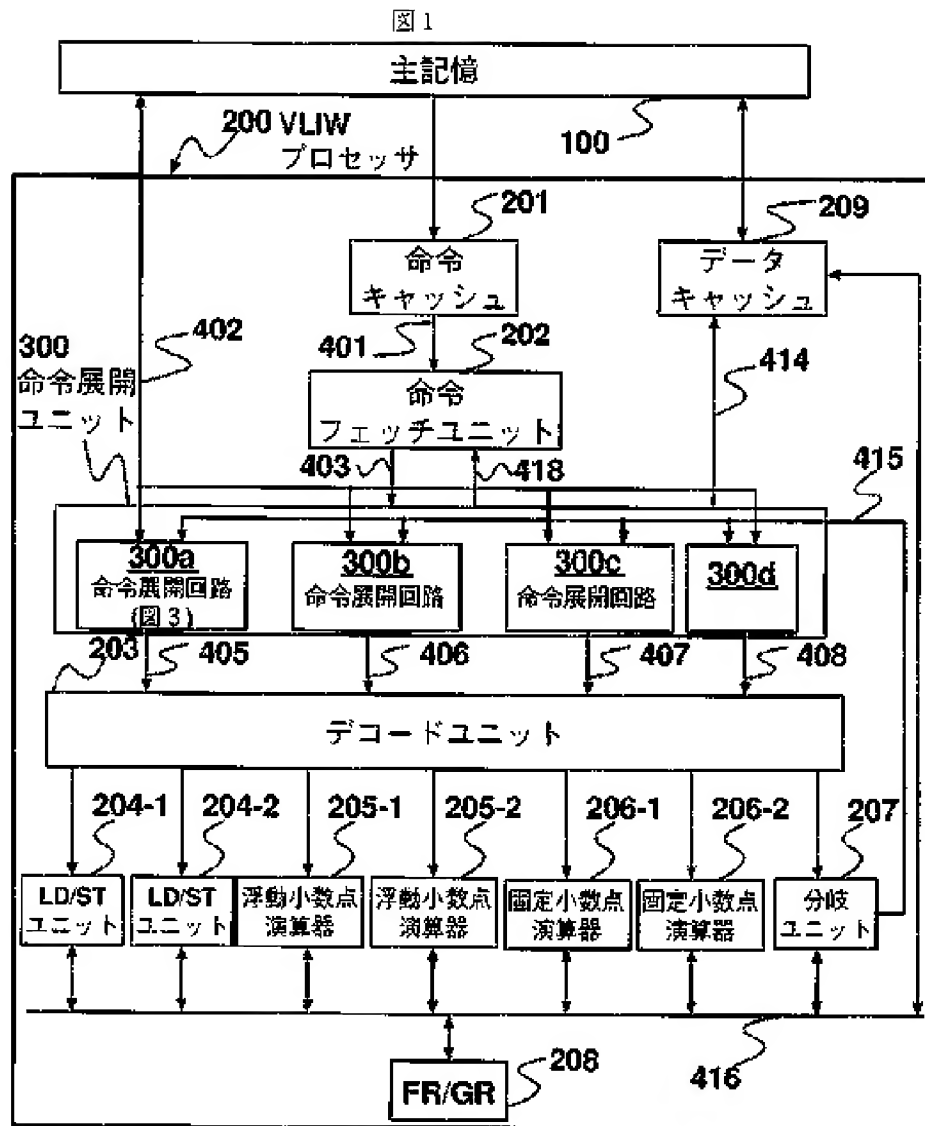
【図15】図12のプログラムのさらに他の一部の実行の様子を説明する図。

【図8】

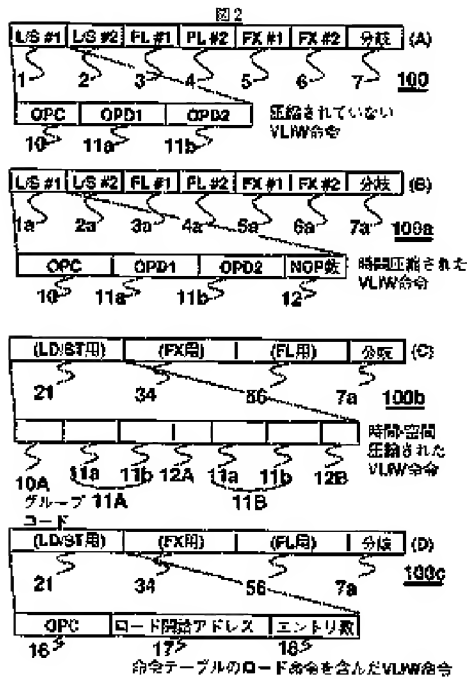
図8

```
DO 1 K=1,400,4
  X(K)=Q+Y(K)*{R*Z(K+10)+T*Z(K+11)}
  X(K+1)=Q+Y(K+1)*{R*Z(K+11)+T*Z(K+12)}
  X(K+2)=Q+Y(K+2)*{R*Z(K+12)+T*Z(K+13)}
  X(K+3)=Q+Y(K+3)*{R*Z(K+13)+T*Z(K+14)}
```

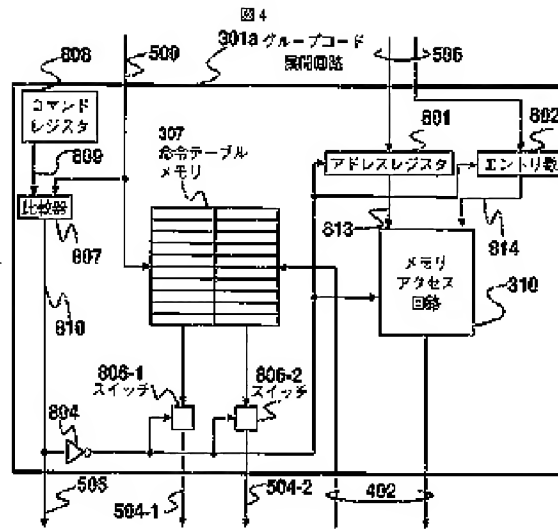
【図1】



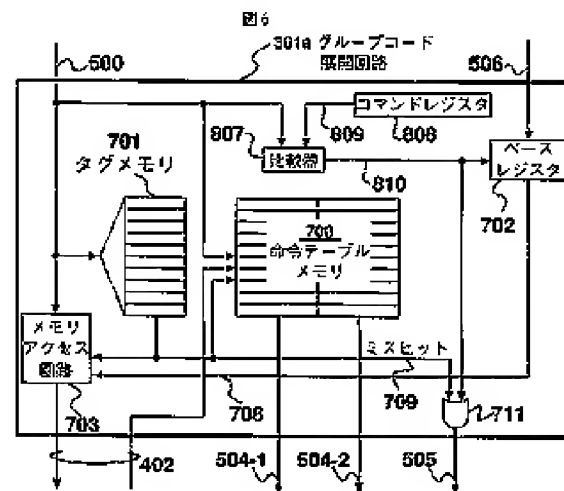
【図2】



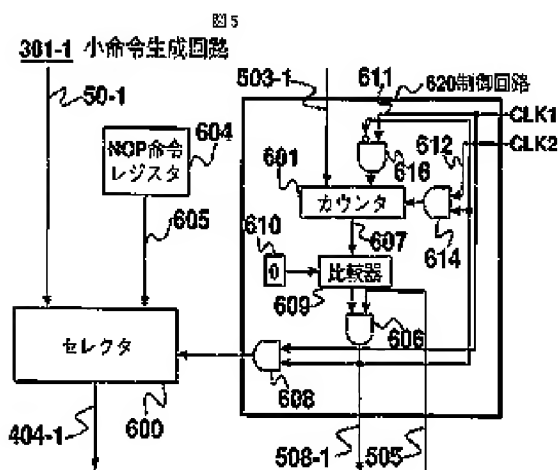
【図4】



【図6】

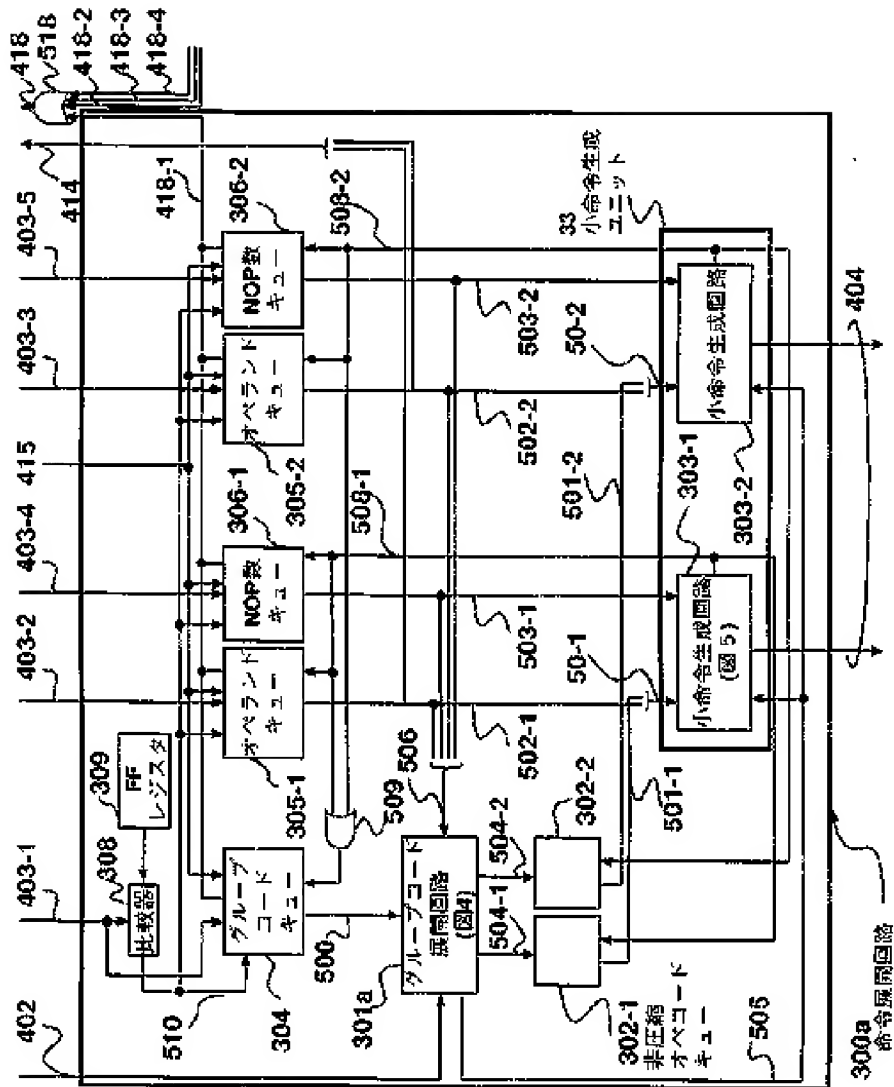


【図5】



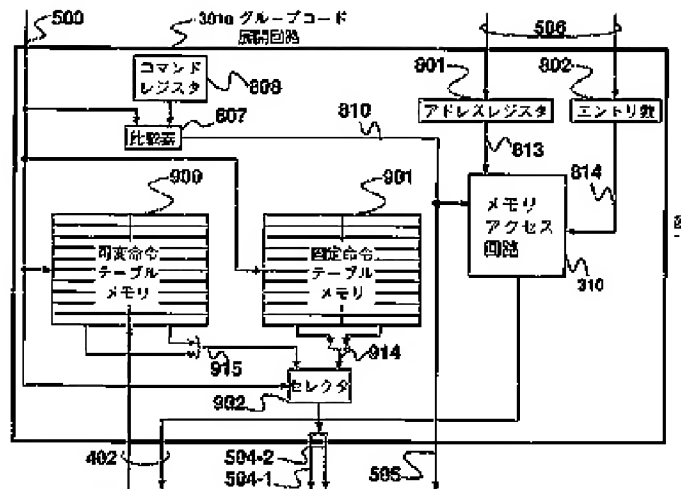
【図3】

図3





【図 7】



【図 9】

図 9

```

(1) LD FR26,2(K+10)
(2) LD FR17
(3) LD FR28
(4) LD FR30
(5) LOOP: MOVE FR4,FR26
(6) LDU FR5,Z(K+11)
(7) FMUL FR6,FR5,FR1
(8) FMUL FR7,FR2,FR4
(9) FADD FR8,FR6,FR7
(10) LDU FR9,Y(K)
(11) FMUL FR10,FR5,FR9
(12) FADD FR11,FR3,FR10
(13) STU FR11,X(5)
(14) LDU FR12,Z(K+12)
(15) FMUL FR13,FR2,FR8
(16) FMUL FR14,FR1,FR12
(17) FADD FR15,FR13,FR14
(18) LDU FR16,Y(K+1)
(19) FMUL FR17,FR15,FR16
(20) FADD FR18,FR3,FR17
(21) STU FR18,X(K+1)
(22) LDU FR19,Z(K+13)
(23) FMUL FR20,FR2,FR12
(24) FMUL FR21,FR1,FR19
(25) FADD FR22,FR20,FR21
(26) LDU FR23,Y(K+2)
(27) FMUL FR24,FR22,FR23
(28) FADD FR25,FR3,FR24
(29) STU FR25,X(K+2)
(30) LDU FR26,Z(K+14)
(31) FMUL FR27,FR2,FR19
(32) FMUL FR28,FR1,FR26
(33) FADD FR29,FR27,FR28
(34) LDU FR30,Y(K+3)
(35) FMUL FR31,FR29,FR30
(36) FADD FR32,FR3,FR31
(37) STU FR32,X(K+3)
(38) BR LOOP

```

【图10】

图 10

LD/ST		浮動小数点演算		固定小数点演算		分岐
LD FR26,Z(K+10)	LD FR1,T	NOP	NOP	NOP	NOP	NOP
LD FR2,R	LD FR3,Q	NOP	NOP	NOP	NOP	NOP
NOP	NOP	NOP	NOP	NOP	NOP	NOP
NOP	NOP	NOP	NOP	NOP	NOP	NOP
NOP	NOP	NOP	NOP	NOP	NOP	NOP
NOP	NOP	NOP	NOP	NOP	NOP	NOP
LDU FR5,Z(K+11)	LDU FR12,Z(K+12)	MOVE FR4,FR26	NOP	NOP	NOP	NOP
LDU FR19,Z(K+13)	LDU FR26,Z(K+14)	NOP	NOP	NOP	NOP	NOP
NOP	NOP	FMUL FR7,FR2,FR4	NOP	NOP	NOP	NOP
NOP	NOP	NOP	NOP	NOP	NOP	NOP
LDU FR9,Y(K)	LDU FR23,Y(K+2)	NOP	NOP	NOP	NOP	NOP
LDU FR16,Y(K+1)	LDU FR20,Y(K+3)	NOP	NOP	NOP	NOP	NOP
NOP	NOP	FMUL FR6,FR1,FR5	FMUL FR13,FR2,FR5	NOP	NOP	NOP
NOP	NOP	FMUL FR14,FR1,FR12	FMUL FR20,FR2,FR12	NOP	NOP	NOP
NOP	NOP	FMUL FR21,FR1,FR19	FADD FR8,FR6,FR7	NOP	NOP	NOP
NOP	NOP	FMUL FR27,FR2,FR19	FADD FR15,FR13,FR14	NOP	NOP	NOP
NOP	NOP	FMUL FR10,FR8,FR9	FMUL FR28,FR1,FR26	NOP	NOP	NOP
NOP	NOP	FMUL FR17,FR15,FR16	FADD FR22,FR20,FR21	NOP	NOP	NOP
NOP	NOP	FADD FR11,FR3,FR10	FADD FR29,FR27,FR28	NOP	NOP	NOP
NOP	NOP	FMUL FR24,FR22,FR23	FADD FR18,FR3,FR17	NOP	NOP	NOP
STU FR11,X(K)	NOP	NOP	FMUL FR31,FR29,FR30	NOP	NOP	NOP
STU FR18,X(K+1)	NOP	FADD FR25,FR3,FR24	NOP	NOP	NOP	NOP
NOP	NOP	NOP	FADD FR32,FR3,FR31	NOP	NOP	NOP
NOP	STU FR25,X(K+2)	NOP	NOP	NOP	NOP	NOP
NOP	STU FR32,X(K+3)	NOP	NOP	NOP	NOP	BR LOOP

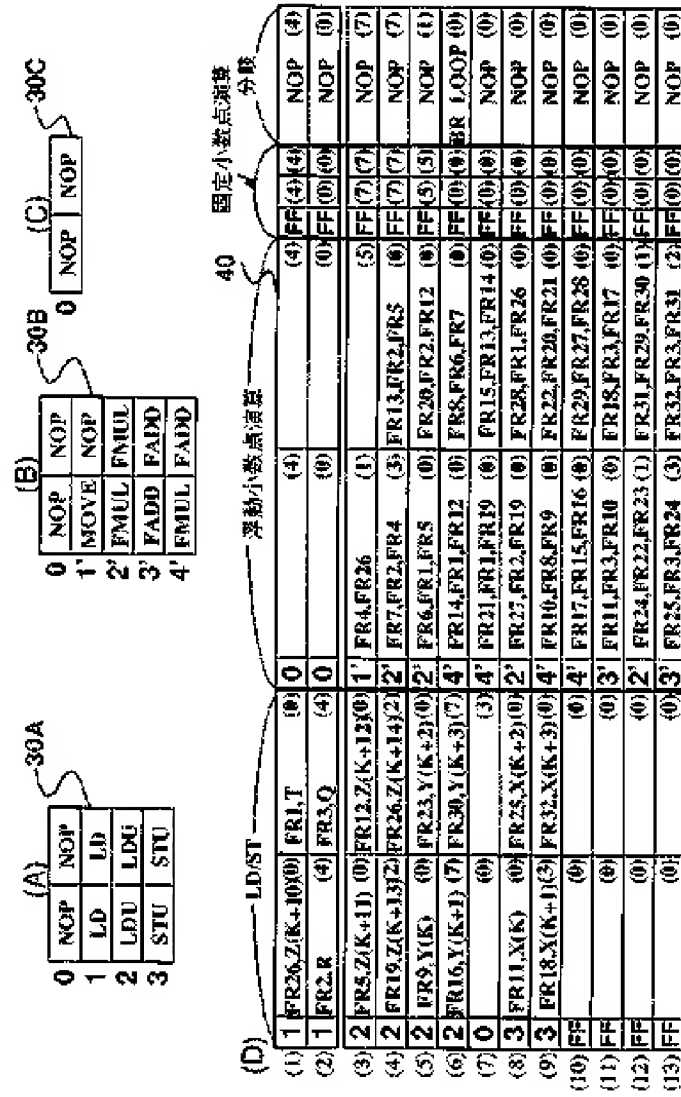
【図11】

図11

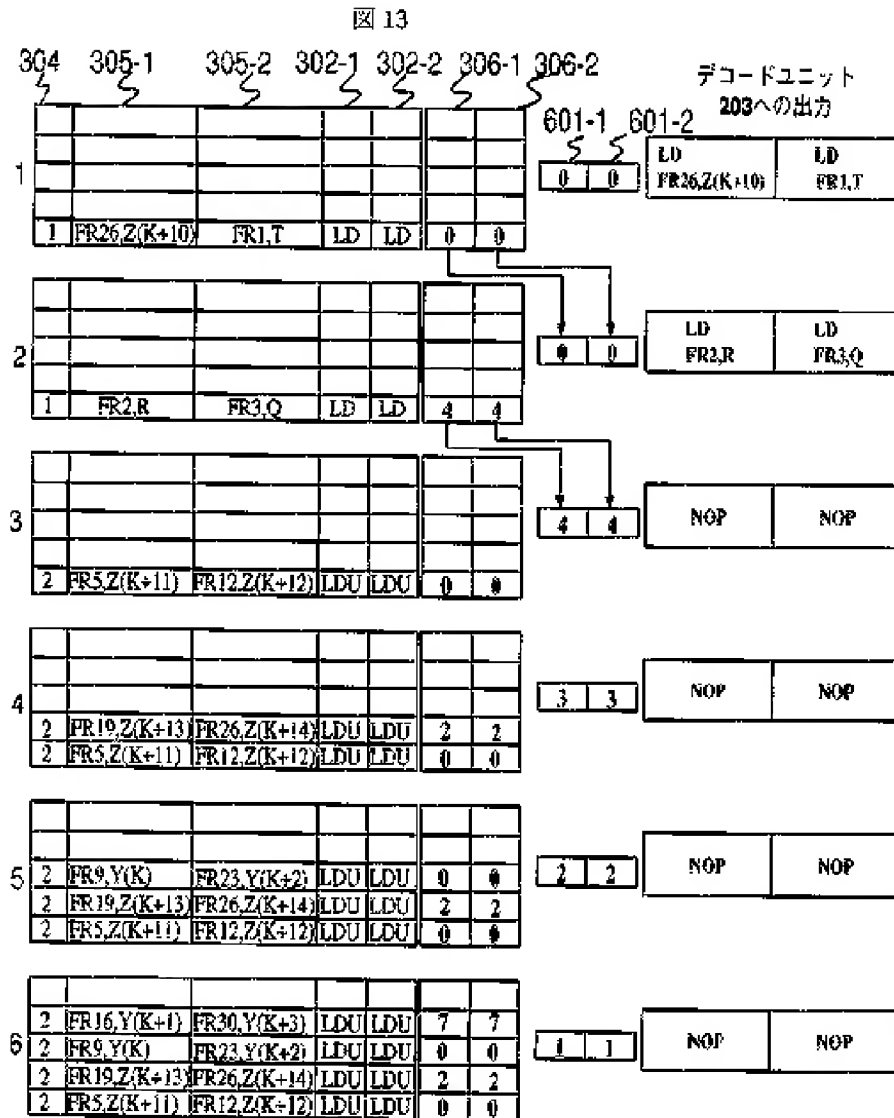
LD/ST			浮動小数点演算			固定小数点演算			分岐		
LD FR26,Z(R+10)	(0)	LD FR1,T	(0)	NOP	(4)	NOP	(0)	NOP(4)	NOP(4)	NOP	(4)
LD FR2,R	(4)	LD FR3,Q	(4)	NOP	(0)	NOP	(0)	NOP(0)	NOP(0)	NOP	(0)
LDU FR5,Z(K+11)	(0)	LDU FR12,Z(K+12)	(0)	MOVE FR4,FR26	(1)	NOP	(5)	NOP(7)	NOP(7)	NOP	(7)
LDU FR10,Z(K+13)	(2)	LDU FR26,Z(K+14)	(2)	FMUL FR7,FR2,FR4	(3)	FMUL FR13,FR2,FR5	(0)	NOP(7)	NOP(7)	NOP	(7)
LDG FR0,Y(K)	(0)	LDU FR23,Y(K+2)	(0)	FMUL FR6,FR1,FR5	(0)	FMUL FR24,FR2,FR12	(0)	NOP(5)	NOP(5)	NOP	(1)
LDU FR16,Y(K+1)	(7)	LDU FR30,Y(K+3)	(7)	FMUL FR14,FR1,FR12	(0)	FADD FR8,FR6,FR7	(0)	NOP(0)	NOP(0)	BR LOOP	(0)
NOP	(0)	NOP	(3)	FMUL FR21,FR1,FR19	(0)	FADD FR15,FR13,FR14	(0)	NOP(0)	NOP(0)	NOP	(0)
STU FR11,X(K)	(0)	STU FR25,X(K+2)	(0)	FMUL FR22,FR2,FR19	(0)	FMUL FR28,FR1,FR26	(0)	NOP(0)	NOP(0)	NOP	(0)
STU FR18,X(K+1)	(3)	STU FR32,X(K+3)	(0)	FMUL FR10,FR5,FR9	(0)	FADD FR12,FR20,FR21	(0)	NOP(0)	NOP(0)	NOP	(0)
NOP	(0)	NOP	(0)	FMUL FR12,FR15,FR16	(0)	FADD FR29,FR27,FR28	(0)	NOP(0)	NOP(0)	NOP	(0)
NOP	(0)	NOP	(0)	FADD FR11,FR3,FR10	(0)	FADD FR18,FR3,FR17	(0)	NOP(0)	NOP(0)	NOP	(0)
NOP	(0)	NOP	(0)	FMUL FR24,FR22,FR23	(1)	FMUL FR31,FR29,FR30	(1)	NOP(0)	NOP(0)	NOP	(0)
NOP	(0)	NOP	(0)	FADD FR23,FR1,FR24	(3)	FADD FR32,FR3,FR31	(2)	NOP(0)	NOP(0)	NOP	(0)

[図12]

図12



【図13】



【図14】

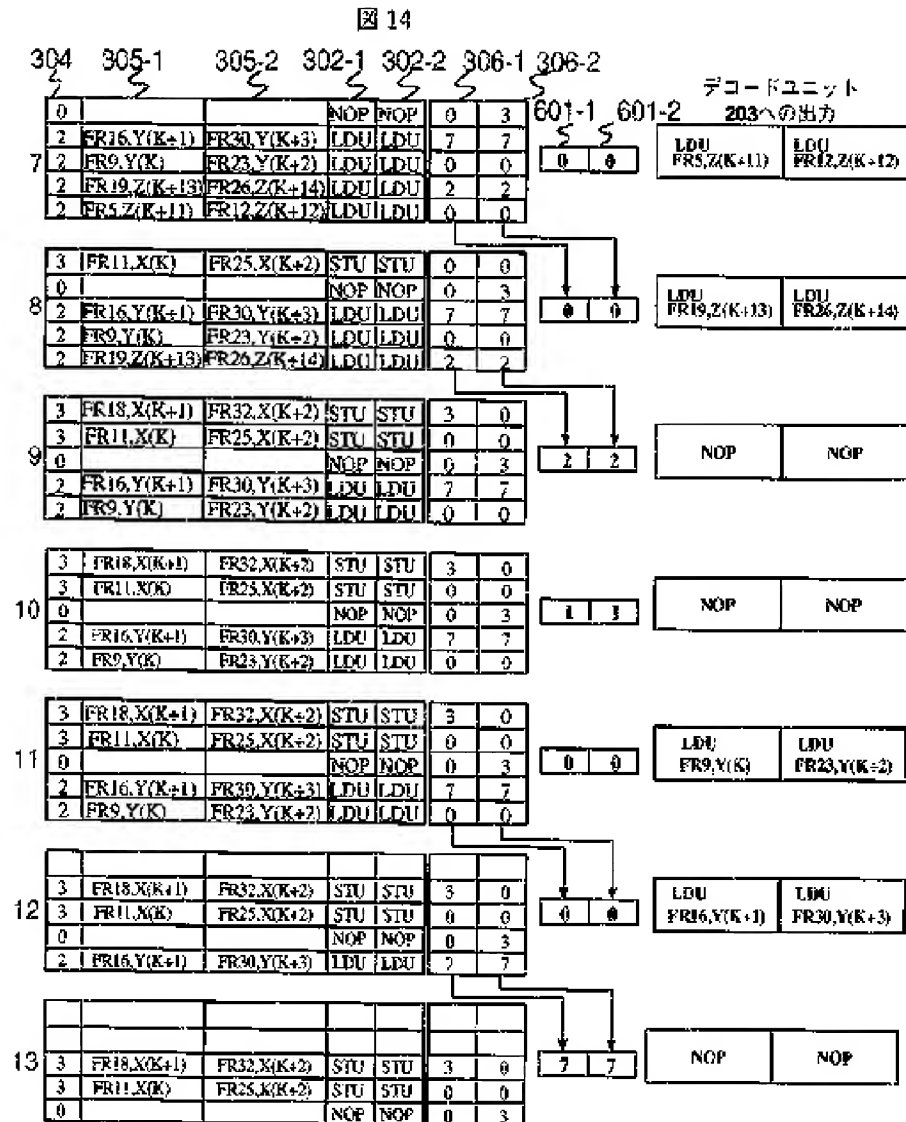
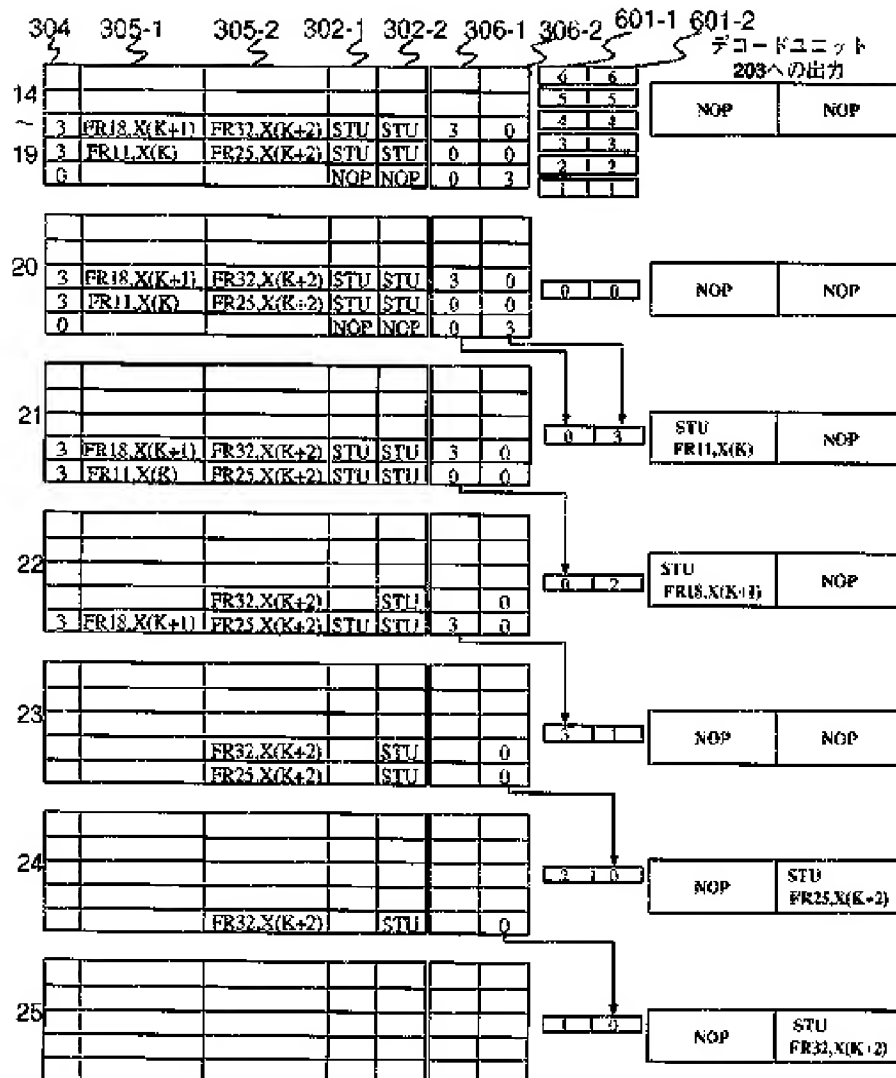


図 15



(72)発明者 戸塚 米太郎  
東京都国分寺市京恋ヶ窪1丁目280番地  
株式会社日立製作所中央研究所内

(72)発明者 長島 重夫  
東京都国分寺市京恋ヶ窪1丁目280番地  
株式会社日立製作所中央研究所内